

Реферат

Пояснительная записка содержит 135 листов, 47 рисунков, 34 таблицы, 10 источников, 3 приложения.

СИСТЕМА, ДАТЧИК, КОНТРОЛЬ, ГАЗОВЫЙ СОСТАВ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, БАЗА ДАННЫХ

Объектом исследования является система контроля газового состава технически сложных объектов, в помещениях которых возможно наличие токсичных и взрывоопасных паров и газов.

Цель работы – разработать для предприятия нефтегазового комплекса программное обеспечение «Программное обеспечение АРМ оператора системы контроля газового состава», являющегося составной частью автоматизированной системы контроля газового состава.

В процессе работы изучена специфика контроля газового состава, изучена система контроля технического оборудования на предприятии нефтегазового комплекса, разработана функциональная модель работы программного обеспечения, разработана структура базы данных.

В результате проделанной работы было разработано программное обеспечение АРМ для контроля технического оборудования на предприятии нефтегазового комплекса.

Среда разработки - Microsoft Visual Studio C# 2008 и СУБД SQL Server 2008.

					ПГУ 1.090301.04.001 ПЗ						
Изм.	Лист	№ докум.	Подпись	Дата	Программное обеспечение АРМ оператора системы контроля газо- вого состава Пояснительная записка			Лит.	Лист	Листов	
Разраб.		Бычинин М.А.									
Провер.		Трокоз Д.А.								4	135
								ФВТ гр. 13ВВ1			
Н. Контр.		Кучин А.В.									
Утв.		Пашенко Д.В.									

Содержание

Обозначения и сокращения.....	7
Введение	8
1 Анализ предметной области	11
2 Требования к разработке приложения.....	19
2.1 Основание для разработки	19
2.2 Назначение разработки	19
2.3 Требования к программе	19
2.3.1 Требования к функциональным характеристикам	19
2.3.2 Требования к надежности	21
2.3.3 Требования к составу и параметрам технических средств.....	21
2.3.4 Требования к информационной и программной совместимости	22
2.4 Требования к программной документации	23
2.5 Стадии и этапы разработки.....	23
2.6 Порядок контроля и приемки	25
3 Функциональное проектирование системы	26
3.1 Общие сведения	26
3.2 Описание функциональной модели	27
4 Инфологическое проектирование	31
4.1 Общие сведения	31
4.2 Логическое проектирование	31
4.3 Физическое проектирование.....	34
4.4 Разработка структуры связей.....	37
4.5 Нормализация базы данных.....	40
5 Анализ целесообразности использования средств, языков и технологий разработки.....	43
6 РЕАЛИЗАЦИЯ	46
6.1 Общие сведения	46

6.2 Функциональное назначение	46
6.3 Входные данные.....	47
6.4 Выходные данные	47
6.5 Описание логической структуры	48
6.7 Описание запросов к системе	71
6.6 Используемые технические средства	74
6.7 Вызов и загрузка	74
7 Программа и методика испытаний	75
7.1 Объект и цель испытаний	75
7.2 Требования к программе.....	75
7.3 Методы испытаний.....	75
8 Описание применения	80
ЗАКЛЮЧЕНИЕ	89
Список использованных источников	90
ПРИЛОЖЕНИЕ А.....	91
ПРИЛОЖЕНИЕ Б	101
ПРИЛОЖЕНИЕ В	111
ПРИЛОЖЕНИЕ Г	134

Обозначения и сокращения

НПВ – нижний предел взрываемости

РЭМ – регистратор электронный многоканальный

АРМ – автоматизированное рабочее место

ПЭВМ – персональная электронно-вычислительная машина

СКГС – система контроля газового состава

ПДК – предельно допустимая концентрация

КЗ – критическое значение

АС – автоматизированная система

ПО – программное обеспечение

БД – база данных

СУБД – система управления базами данных

					ПГУ 1.090301.04.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

Система нефтепродуктообеспечения играет важную роль в развитии нашей страны. Промышленность, транспорт, сельское и коммунальное хозяйство, население не обходятся без нефтепродуктов — различного вида топлива, газа, масел, смазок, растворителей. Их поставки потребителям осуществляются через широкую сеть нефтебаз и автозаправочных станций (АЗС).

Современная нефтебаза — это сложное и многообразное хозяйство. Оно включает резервуарные парки, разветвленные трубопроводные коммуникации, мощное насосно-силовое оборудование, разнообразные сливоналивные устройства и др. Технически правильная, рациональная эксплуатация объектов нефтебаз возможна только хорошо подготовленными специалистами, имеющими четкое и ясное представление о применяемых технике и технологиях. Актуальной задачей является предотвращение потерь нефтепродуктов от разлива, испарений, загрязнения и т. П.

Всюду, где осуществляется хранение, разлив, переработка и транспортировка горючих и токсичных жидкостей или газов, достаточно высока вероятность возникновения аварийной ситуации. Правильное планирование и размещение систем контроля газового состава (СКГС) в значительной степени уменьшает риск для человека и окружающей среды. На предприятиях нефтегазовой промышленности утечка сжатых газов из трубопроводов, просачивание паров горючих жидкостей через поврежденные вентили или негерметичные соединения может привести к возникновению опасных ситуаций:

а) при утечках горючих веществ воспламенение и взрыв могут привести к пожару и травмам персонала, а также к значительным повреждениям оборудования и сооружений;

б) При утечке токсичных веществ возникает угроза здоровью человека, вызывающая последствия вторичного характера, и возможны человеческие жертвы, то есть тяжелейшие последствия, с трудом поддающиеся оценке.

Для максимального снижения риска возникновения чрезвычайной ситуации при работе с газами и парами токсичных и взрывоопасных веществ эффективным средством являются системы контроля газового состава при условии их правильного размещения и эксплуатации, а также регулярного проведения технического осмотра и функциональной проверки. Правильно используемые, такие системы активно предупреждают о формировании взрывоопасной атмосферы и принимают эффективные меры по устранению аварийной ситуации.

При планировании систем контроля газового состава необходимо иметь конкретный план действий, необходимых для устранения опасной ситуации, которые должны предприниматься при срабатывании системы. Такими действиями могут быть: закрытие газовых вентилей и включение интенсивной вентиляции, отключение производственного электрооборудования. Кроме того, такие действия должны сопровождаться определенными организационными мероприятиями, описанными в планах устранения аварийных ситуаций. Например, в случае активизации сигнализации правилами может предписываться надеть респираторное оборудование и покинуть область поражения, а также использовать переносные газосигнализаторы при возвращении в пораженную область после устранения газа. Главные химические тревоги нарушают бесперебойную работу производства и по этой причине должны возникать как можно реже. Результаты мер, принятых уже после запуска предварительной тревоги, должны быть настолько эффективными, чтобы не возникало необходимости активизировать главную тревогу, нарушающую производственный процесс. Кроме того, системы контроля газового состава должны быть в высшей степени надежными и постоянно находиться в готовности, поскольку, с одной стороны, при срабатывании системы крайне сложно отличить реальную тревогу от ложной, а с другой стороны, даже кратковременные аварии часто приводят к таким же последствиям, что и главные тревоги. Существующие системы контроля газового состава являются совокупностью датчиков подключенных к аварийной сигнализации и не являются интеллек-

					ПГУ 1.090301.04.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

туальными. Таким образом, создание современной системы контроля газового состава технически сложных объектов является актуальной задачей.

					ПГУ 1.090301.04.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ предметной области

В состав нефтебаз и автозаправочных станций входят технически сложные объекты, в помещениях которых возможно наличие токсичных и взрывоопасных паров и газов. При утечках горючих веществ возможное воспламенение и взрыв могут привести к значительным повреждениям оборудования, а также зачастую к пожару и травмам персонала. При утечке токсичных веществ возникает угроза здоровью человека, часто связанная с последствиями вторичного характера, и возможны человеческие жертвы, то есть тяжелейшие последствия, с трудом поддающиеся оценке. В помещениях технически сложных объектов могут использоваться газы, которые при утечке вытесняют воздух из помещения, что приводит к снижению концентрации кислорода ниже предельного значения. Также возможно натекание кислорода выше, чем содержится в воздухе, что также опасно с точки зрения пожарной безопасности.

В целях предотвращения взрывоопасных и критических ситуаций, для обеспечения сохранности дорогостоящего оборудования и сооружений, выполнения требований экологической и эксплуатационной безопасности для обслуживающего персонала необходимо отслеживать концентрацию кислорода, наличие токсичных газов, паров окислителей и горючего.

Контролируемые помещения подразделяются на следующие группы:

- а) I группа – помещения, где необходимо контролировать концентрацию паров окислителя;
- б) II группа – помещения, где необходимо контролировать концентрацию паров горючего;
- в) III группа – помещения, где необходимо контролировать наличие токсичных газов;
- г) IV группа – помещения, где необходимо контролировать концентрацию кислорода.

При планировании систем контроля газового состава необходимо уделить внимание трем ключевым моментам:

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

а) Каковы цели контроля газового состава, какие газы должны контролироваться, каковы предположительные концентрации, объемы и частота утечек?

б) Какой принцип обнаружения газа применен в датчиках, сколько датчиков необходимо, где и как они должны размещаться и калиброваться?

в) Каковы должны быть пороги тревоги и как реагировать на информацию о тревогах[1].

Ответ на первый вопрос описывает реальную задачу, проясняет производственные условия и поставленные цели: должны ли сенсоры выявлять токсичные вещества, или газы, поражающие посредством дефицита кислорода, для защиты человека, или же обнаруживать горючие газы и пары для защиты от взрыва (пожара).

Методы измерения газового состава:

- а) электрохимический принцип;
- б) термокаталитический принцип;
- в) инфракрасный принцип.

Электрохимический принцип особенно эффективен при обнаружении токсичных газов в диапазоне низких концентраций.

Термокаталитический и инфракрасный принципы пригодны для обнаружения горючих веществ при концентрациях до нижнего предела взрываемости. В зависимости от конкретного приложения, каждый принцип может обладать как достоинствами, так и недостатками, поэтому производитель при поддержке своей лаборатории должен дать достоверный ответ о выборе наиболее надежного метода для решения данной задачи.

Второй ключевой момент включает самый типичный вопрос: «Сколько квадратных метров может контролировать один сенсор и как их размещать?» К сожалению, технические правила не могут дать на него конкретный ответ – данная проблема целиком зависит от реальной ситуации и поставленной задачи.

Различают три основные стратегии размещения сенсоров – естественно, возможны компромиссы и плавные переходы от одного подхода к другому:

а) точечный мониторинг (spot monitoring): потенциальные источники утечек (например, вентили, краны, муфты, фланцы, сильфоны), а также места их расположения известны. поэтому датчики можно расположить так, чтобы надежно и своевременно выявлять места утечек;

б) площадной мониторинг (area monitoring): потенциальные источники утечки не локализируются, но располагаются в определенной области, пространстве. датчики должны распределяться по всей области;

в) периферийный мониторинг (fence monitoring): потенциальные источники утечек не поддаются локализации, поэтому мониторинг производится на внешней границе участка, для обнаружения превышения концентрации газов в ограниченной опасной зоне [1].

При этом отчетливо просматривается градация концепций безопасности. Если при точечном мониторинге можно целенаправленно использовать небольшое количество датчиков, то площадной мониторинг можно реализовать лишь при наличии большого количества датчиков. Подобный мониторинг используется, например, при контроле за хранением горючих жидкостей, причем датчики распределяются равномерно по всей площади.

Обычно в помещениях устанавливается до 12 точек контроля присутствия токсичных газов, паров окислителя и горючего и от 1 до 4 точек контроля кислорода.

Датчики газового состава устанавливаются в местах возможного появления токсичных и взрывоопасных паров, а вторичная аппаратура устанавливается в чистой зоне, где данные пары отсутствуют. При размещении датчиков необходимо учитывать определенные правила. Пары горючих жидкостей всегда тяжелее воздуха. Они растекаются на уровне пола. Поэтому датчики должны размещаться как можно ниже над полом, но в доступном для калибровки месте.

Лишь три горючих газа значительно легче воздуха – это метан, аммиак и водород. Не находясь в жидком состоянии (при очень низких температурах), они поднимаются вверх, собираясь и скапливаясь под потолком. Утечки этих трех газов на открытом воздухе не представляют больших проблем. Токсичные газы, которые хоть и тяжелее воздуха, но обычно представлены в воздухе лишь низкими концентрациями, должны обнаруживаться на уровне головы (в зоне дыхания), так как их распространение в значительной мере определяется воздействием конвекции – нисходящих и восходящих потоков воздуха, образующихся вследствие неравномерного нагрева поверхности.

При разработке систем контроля газового состава необходимо иметь конкретный план действий, необходимых для устранения опасной ситуации, которые должны предприниматься при срабатывании системы. Так, при превышении определенной, предварительно установленной концентрации газа (например, 10% от нижнего предела взрываемости, НПВ) система может закрыть газовые вентили и включить интенсивную вентиляцию. Если же этих действий окажется недостаточно и концентрация газа в контролируемой области продолжит расти, не смотря на принимаемые меры, то при превышении второго порога тревоги (например, 30% НПВ), принимаются принудительные меры – отключаются все потенциальные источники искрения (например, промышленное электрооборудование), осуществляется промывка трубопроводов, охлаждение горячих поверхностей и активизация водных завес. Кроме того, такие действия должны сопровождаться определенными организационными мероприятиями, описанными в планах устранения аварийных ситуаций. Например, при утечке токсичных газов в случае активизации главной тревоги сигнализации предписывается надеть респираторное оборудование и покинуть область поражения, а также использовать переносные газосигнализаторы при возвращении в пораженную область после устранения газа.

Оптимальная структура системы контроля газового состава представлена на рисунке 1.1.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

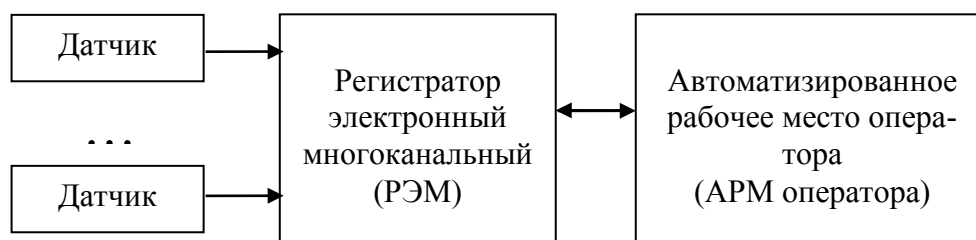


Рисунок 1.1 – Структура интеллектуальной системы мониторинга

Структурная схема системы контроля газового состава представлена на рисунке А.1.

Состав системы контроля газового состава:

- 16 датчиков;
- контроллер;
- регистратор электронный многоканальный;
- источник питания контроллера и регистратора;
- блок световой и звуковой сигнализации;
- автоматизированное рабочее место оператора.

Состав автоматизированного рабочего места оператора:

- сервер;
- ПЭВМ;
- монитор;
- клавиатура;
- манипулятор типа «мышь»;
- принтер;
- источник бесперебойного питания;
- панель отображения.

Ключевым элементом любой системы контроля газового состава являются чувствительные элементы (сенсоры). От них зависит не только надежность измерений, но и эксплуатационные издержки всей системы. В настоящее время используются электрохимические, каталитические и инфракрасные сенсоры.

Современные датчики, специально разработаны для стационарных систем контроля газового состава, обладают встроенной памятью, которая содержит информацию о газах, диапазоне измерения, перекрестной чувствительности, температурной компенсации и времени наработки сенсора, для самодиагностики и корректировки показаний.

Датчики передают свои показания контроллеру, который преобразует данные сигналы в сигналы понятные ПЭВМ. Контроллер объединяет преобразованные сигналы и передает их по интерфейсу RS-232, RS 485 регистратору.

Для автономного сбора, обработки, визуализации и хранения информации используется регистратор электронный многоканальный.

Функции регистратора СКГС:

- а) получение необходимой, полной и достоверной информации о параметрах газового состава в реальном масштабе времени;
- б) формирование математических каналов;
- в) выдача управляющих сигналов для сигнализации и управления;
- г) самодиагностика и перестройка диапазона измерений;
- д) адаптация к изменению окружающей среды;
- е) контроль, внешняя сигнализация и регулирование при превышении установленных значений измеряемых величин;
- ж) регистрация, отображение и архивирование результатов измерения и системных сообщений;
- и) представление результатов измерения в аналоговом и цифровом виде и отображение на дисплее.

Регистратор позволяет сохранять и загружать свои настройки на сервер интерфейсу RS-232, RS 485, Ethernet.

Одновременно с передачей данных на регистратор контроллер передает по интерфейсу Ethernet данные на сервер. Сервер хранит полученную информацию и расширяет функции регистратора по ее обработке, производит расчет математических моделей (например, расчет последствий чрезвычайной ситуа-

ции). ПЭВМ считывает данные с сервера, отображает полученные данные на экране монитора.

При отсутствии связи контроллера с сервером, показания датчиков сохраняются в базе данных регистратора, а после восстановления работы сервера, сохраненные показания можно скопировать на сервер.

При возникновении аварийной ситуации сервер посылает сигнал контроллеру для включения аварийной световой и звуковой сигнализации. Если сервер не работает, управляющий сигнал подает регистратор.

Возможные области применения:

- а) сооружения наземной космической инфраструктуры;
- б) нефтегазовая промышленность;
- в) химическая промышленность;
- г) топливно-энергетический комплекс.

При реконструкции и введении в систему датчиков другого назначения она может быть использована в различных отраслях промышленности.

Система мониторинга обеспечивает:

- а) измерение концентрации паров окислителя и горючего в помещениях I и II групп;
- б) измерение концентрации токсичных газов в помещениях III группы;
- в) измерение объемной доли содержания кислорода в воздухе помещений IV группы в диапазоне от 15 % до 25 %;
- г) анализ результатов измерения, формирование и выдачу на экран монитора пэвм и устройства сигнализации:

1) «НОРМА» при нахождении контролируемых параметров в допустимых пределах (концентрация паров окислителя и горючего ниже 1 ПДК и содержание кислорода от 19% до 23%);

2) «НЕНОРМА» при показаниях хотя бы одного из преобразователей контролируемого помещения значений равных или превышающих 10 ПДК;

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

З) «НЕНОРМА К» при показаниях хотя бы одного из датчиков контролируемого помещения значений объемной доли содержания кислорода, равной или ниже 19 % и более 23 %;

д) автоматическое выключение сигнала «норма» в случае неисправности всех измерительных преобразователей в контролируемом помещении;

е) автоматическое отключение звуковой сигнализации в контролируемых помещениях по истечении 60 с. после ее включения;

ж) возможность принудительного выключения звуковой сигнализации в контролируемом помещении оператором посредством удаленной пэвм и (или) с помощью выключателя из сооружения;

и) бесперебойное питание на 10-20 минут в случае отключения от системы энергосбережения для сохранения информации на ПЭВМ и корректного завершения работы и перезапуска системы (после восстановления электропитания).

Существующие системы контроля газового состава являются совокупностью датчиков, подключенных к аварийной сигнализации, и не являются интеллектуальными. Для интеллектуализации системы контроля газового состава необходимо ее подключение к ПЭВМ. Для взаимодействия системы и оператора необходимо разработать программное обеспечение автоматизированного рабочего места оператора, позволяющего визуально отследить состояние объекта и принять необходимые меры для предотвращения чрезвычайных ситуаций.

2 Требования к разработке приложения

2.1 Основание для разработки

Разработка ведется на основе задания, выданное руководителем бакалаврской работы и утвержденное заведующим кафедрой ВТ.

2.2 Назначение разработки

Программное обеспечение АРМ оператора системы контроля газового состава предназначено для хранения данных об используемых датчиках и чувствительных элементах на технически сложных объектах, примененных каналах связи, мониторинга газового состава помещений данных объектов, хранения показаний датчиков и сигнализации при приближении чрезвычайных ситуаций.

2.3 Требования к программе

2.3.1 Требования к функциональным характеристикам

Функции программного обеспечения АРМ оператора системы контроля газового состава:

а) имитация сигналов с датчика (с помощью псевдослучайной последовательности), с перспективной заменой на прием сигналов с многоканального электронного регистратора;

б) контроль 16 измерительных каналов;

в) хранение данных о примененных датчиках и чувствительных элементах каждого из каналов;

г) хранение информации об объектах, на которых установлены датчики и вида примененных каналов связи для соединения с арм;

д) формирование 6 математических каналов для расчета необходимых параметров из отслеживаемых;

					ПГУ 1.090301.04.001 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

е) установка режима нормальной работы для каждого измерительного канала:

- 1) минимальное предельно допустимое значение (ПДК min);
- 2) максимальное предельно допустимое значение (ПДК max);
- 3) минимальное критическое значение (КЗ min);
- 4) максимальное критическое значение (КЗ max);

ж) отслеживание состояния каждого измерительного канала в виде:

- 1) графиков;
- 2) цифровых показаний (таблиц);
- 3) шкал;
- 4) вертикальных столбиковых диаграмм;

Цикл отображения (временной интервал вывода каждой следующей точки на экран) задаётся пользователем при настройке из следующего ряда значений: 1с, 2 с, 5 с, 10 с, 30 с, 1 мин, 2 мин, 5 мин.

и) Сигнализация в случае выхода за пределы нормального режима работы и вывод вариантов решения возникшей проблемы:

- 1) изменить входное значение канала (при работе имитаторов датчиков);
- 2) изменить ограничения нормального режима работы;
- 3) включить вентиляцию;
- 4) перекрыть кран подачи газа;
- 5) отследить показания измерительного канала;
- 6) просмотреть последние действия пользователей системы;

к) сохранение значений измерительных каналов в базе данных;

л) протоколирование действий пользователя и сообщений системы;

м) вывод архивной информации.

Входными данными системы должны быть:

- а) показания датчиков (генератор случайных чисел, в перспективе показания с регистратора);
- б) информация об объектах (название, адрес, описание);
- в) информация о датчиках (название, описание);
- г) информация о чувствительных элементах (название, описание);
- д) информация о каналах связи;
- е) информация об измерительных каналах (ПДК min, ПДК max, КЗ min, КЗ max);
- ж) информация о математических каналах (функция для расчета, ПДК min, ПДК max, КЗ min, КЗ max);
- и) информация о пользователях (фамилия, имя, отчество, дата рождения, должность);
- к) дата и время.

Выходными данными системы должны быть:

- а) отчет об объектах и датчиках;
- б) отчет о состоянии каналов;
- в) журнал действий пользователей;
- г) журнал сообщений системы.

2.3.2 Требования к надежности

Информационная система должна выдавать сообщения о возникающих ошибках при неверном задании исходных данных, поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей.

2.3.3 Требования к составу и параметрам технических средств

2.3.3.1 Состав автоматизированного рабочего места

Системный блок с характеристиками:

- а) Процессор Pentium;

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

- б) Оперативная память объемом 2Гбайт;
- в) Жесткий диск емкостью 500 Гбайт;
- г) Сетевая карта Ethernet 100/1000 Мбит/с;
- а) Монитор тип – жидкокристаллический;
- б) Интерфейсы (входы) – VGA (D-Sub), DVI;

2.3.3.2 Состав серверов пункта мониторинга

— Системный блок с характеристиками не хуже:

- а) Процессор типа Intel Xeon с тактовой частотой 3,2 ГГц;
- б) Оперативная память 4 Гбайт;
- в) 2 каналный контроллер с RAID массивом;
- г) Объем дисковой памяти RAID массива 4x500 Гбайт;
- д) Сетевая карта 100/1000 Мбит/с;
- е) Оптический накопитель DVD – RW.

2.3.4 Требования к информационной и программной совместимости

Общее программное обеспечение должно включать в себя версии, не ниже указанных:

- а) серверная операционная система: Windows 2008 Server R2;
- б) операционная система для рабочих мест: Windows XP/7;
- в) программное обеспечение системы управления базами данных SQL Server 2008;
- г) средство проектирования программного обеспечения Sybase Powerdesigner 15;
- д) программная среда разработки приложений Microsoft Visual Studio C# 2008;
- е) программное обеспечение антивирусной защиты.

2.4 Требования к программной документации

Комплект разрабатываемых документов должен включать:

- а) техническое задание;
- б) текст программы (листинг);
- в) описание программы;
- г) программу и методику испытаний;
- д) описание применения.

Программная документация должна разрабатываться в соответствии с требованиями ЕСПД.

2.5 Стадии и этапы разработки

Процесс разработки информационной системы должен соответствовать стадиям и этапам разработки, указанным в таблице 2.1.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

Таблица 2.1 – Стадии и этапы разработки

Стадии	Этапы работ
1 Формирование требований к АС	1.1 Обследование объекта и обоснование необходимости создания АС
	1.2 Формирование требований пользователя к АС
	1.3 Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания)
2 Разработка концепции АС	2.1 Изучение объекта
	2.2 Проведение необходимых научно-исследовательских работ
	2.3 Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя
	2.4 Оформление отчёта о выполненной работе
3 Техническое задание	Разработка и утверждение технического задания на создание АС
4 Эскизный проект	4.1 Разработка предварительных проектных решений по системе и её частям
	4.2 Разработка документации на АС и её части
5 Технический проект	5.1 Разработка проектных решений по АС
	5.2 Разработка документации на АС и её части
	5.3 Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку
	5.4 Разработка заданий на проектирование в смежных частях проекта объекта автоматизации
6 Рабочая документация	6.1 Разработка рабочей документации на систему и её части
	6.2 Разработка или адаптация программ
7 Ввод в действие	7.1 Подготовка объекта автоматизации к вводу АС в действие
	7.2 Подготовка персонала
	7.3 Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)
	7.4 Строительно-монтажные работы
	7.5 Пусконаладочные работы
	7.6 Проведение предварительных испытаний
	7.7 Проведение опытной эксплуатации
	7.8 Проведение приёмочных испытаний
8 Сопровождение АС	8.1 Выполнение работ в соответствии с гарантийными обязательствами
	8.2 Послегарантийное обслуживание

2.6 Порядок контроля и приемки

В целях контроля правильности решения задачи и работы программы, необходимо проводить тестирование, в котором должны быть учтены все возможные варианты исходных данных, отражающие работу всех частей программы. Для тестирования программы должны быть подготовлены тестовые примеры, охватывающие все функциональные возможности программы. Пояснительная записка должна соответствовать госту. Приемка осуществляется заказчиком в соответствии с правильностью работы программы, определяемой по результатам выполнения тестовых примеров.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

3 Функциональное проектирование системы

3.1 Общие сведения

При проектировании информационных систем применяют структурный подход (функционально-модульный). В его основе лежат принципы функциональной декомпозиции. Это означает, что система описывается в терминах иерархии функций системы и передачи информации между функциями.

Процесс моделирования системы начинается с определением контекста, т.е. Наиболее абстрактного уровня описания системы в целом. В контекст входит определение субъекта моделирования, цели и точки зрения на модель.

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами. На определение субъекта системы будет существенно влиять позиция, с которой рассматривается система, и цель моделирования – вопросы, на которые построенная модель должна дать ответ.

Существуют три методологии создания моделей процессов: IDEF0 (функциональная модель), IDEF3 (workflow Diagram), DFD (dataflow Diagram), каждая из которых решает свои специфические задачи.

В нотации IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации. Под моделью в IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы.

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ.

					ПГУ 1.090301.04.001 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. DFD описывает:

- а) Функции обработки информации (работы);
- б) документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- в) внешние ссылки (external references) , которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- г) таблицы для хранения документов (хранилище данных, data store).

Для описания логики взаимодействия информационных потоков более подходит IDEF3, называемая также workflow diagramming - методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. Диаграммы Workflow могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

3.2 Описание функциональной модели

Разработка функциональной модели производилась с помощью CASE - средства Sybase Powerdesigner 15.3.

Для моделирования системы в качестве контекстной диаграммы использовалась методология DFD (Data Flow Diagram). Целью методики является построение модели рассматриваемой системы в виде диаграммы потоков данных. Диаграммы потоков данных предназначены, прежде всего, для описания документооборота и обработки информации. Вершиной древовидной

					ПГУ 1.090301.04.001 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

структуры диаграмм является контекстная диаграмма, представленная на рисунке А.1 и представляющая собой самое общее описание системы и ее взаимодействие с внешней средой.

На данной диаграмме, построенной в методологии DFD, изображён функциональный блок «Контроль газового состава».

Внешними сущностями являются:

- а) контроллер;
- б) администратор;
- в) оператор
- г) источник даты и времени.

Входными потоками данных для данной системы являются:

- а) информация о пользователе (табельный номер, фамилия, имя, отчество, дата рождения, должность);
- б) информация об объектах (адрес, координаты, описание, установленные на объект датчики);
- в) информация о датчиках (чувствительный элемент, название, описание, используемый канал связи, координаты, номер выделенного канала);
- г) информация о чувствительных элементах (название, описание);
- д) информация о каналах связи (название, описание);
- е) информация об измерительных каналах (название, единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max);
- ж) информация о математических каналах (название, функция, единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max);
- и) показания датчиков (величина измеряемого параметра);
- к) дата и время.

Выходными данными являются:

- а) отчет об объектах и датчиках;
- б) отчет о состоянии объектов;
- в) журнал действий пользователей;
- г) журнал сообщений системы.

					ПГУ 1.090301.04.001 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

На основе анализа предметной области осуществлена декомпозиция системы.

Декомпозиция представлена процессами: «учет и анализ показаний датчиков», «учет и анализ действий пользователя», «формирование отчета о состоянии объектов», «формирование журнала сообщений системы», «формирование отчета об объектах и датчиках», «формирование журнала действий пользователя». Диаграмма приведена на рисунке А.2.

Система имеет входные потоки данных:

- а) показания датчиков;
- б) дата и время;
- в) информация о пользователе;
- г) информация об объектах;
- д) информация о датчиках;
- е) информация о каналах связи;
- ж) информация об измерительных каналах;
- и) информация о математических каналах;
- к) информация о чувствительных элементах.

У системы имеется четыре выходных потока данных:

- а) отчет об объектах и датчиках;
- б) отчет о состоянии объектов;
- в) журнал действий пользователей;
- г) журнал сообщений системы.

Процессы «формирование отчета о состоянии объектов», «формирование журнала сообщений системы», «формирование отчета об объектах и датчиках», «формирование журнала действий пользователя» являются элементарными, а процессы «учет и анализ показаний датчиков» и «учет и анализ действий пользователя» имеют сложную структуру и требуют декомпозиции.

Диаграмма декомпозиции процесса «учет и анализ показаний датчиков» (рисунок А.3) представлена процессами: «учет показаний датчиков», «вычисление математических каналов», «анализ состояния каналов», «формирование

					ПГУ 1.090301.04.001 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

сообщений системы». Дата и время, показания датчиков и данные математических каналов регистрируются в хранилище «состояние», сообщения системы регистрируются в хранилище «сообщения системы», данные об измерительных каналах считываются из хранилища «измерительный канал», данные о математических каналах считываются из хранилища «математический канал».

Диаграмма декомпозиции процесса «учет и анализ действий пользователя» (рисунок А.4) представлена процессами: «корректировка информации о пользователе», «корректировка информации о чувствительных элементах», «корректировка информации о датчиках», «корректировка информации об объектах», «корректировка информации о каналах связи», «корректировка информации об измерительных каналах», «корректировка информации о математических каналах», «анализ действий пользователя», «учет действий пользователя». Данные о пользователе сохраняются в хранилище «пользователь», данные о чувствительных элементах - в хранилище «чувствительный элемент», данные о датчиках - в хранилище «датчик», данные об объектах - в хранилище «объект», данные о каналах связи - в хранилище «канал связи», данные об измерительных каналах - в хранилище «измерительный канал», данные о математических каналах - в хранилище «математический канал», данные о действиях пользователя - в хранилище «действия пользователя». Данные после корректировки сравниваются с данными из хранилищ «пользователь», «чувствительный элемент», «датчик», «объект», «канал связи», «измерительный канал», «математический канал» формируя «журнал действия пользователя». Данные о чувствительных элементах, датчиках, объектах, каналах связи, измерительных каналах считываются из хранилищ «чувствительный элемент», «датчик», «объект», «канал связи», «измерительный канал» соответственно.

Процессы, выделенные в результате декомпозиции, являются элементарными и не требуют дальнейшей декомпозиции. Диаграммы представлены на рисунках А.1 – А.4.

					ПГУ 1.090301.04.001 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

4 Инфологическое проектирование

4.1 Общие сведения

На этапе инфологического проектирования базы данных должна быть построена модель предметной области, не привязанная к конкретной СУБД, понятная не только разработчикам информационной системы, но и экономистам, менеджерам и другим специалистам. В то же время модель предметной области должна максимально точно отражать семантику предметной области, выявлять бизнес-правила и позволять легко перейти к модели данных конкретной СУБД [2].

Результатом инфологического проектирования является ER-диаграмма ("сущность-связь"). Графический подход к созданию моделей значительно упрощает процесс разработки [3].

Модель Сущность-Связь (ER-модель) (англ. Entity-relationship model (ERM) или англ. Entity-relationship diagram (ERD)) — модель данных, позволяющая описывать концептуальные схемы. Представляет собой графическую нотацию, основанную на блоках и соединяющих их линиях, с помощью которых можно описывать объекты и отношения между ними какой-либо другой модели данных. В этом смысле ER-модель является метамоделью данных, то есть средством описания моделей данных [3].

ER-модель удобна при прототипировании (проектировании) информационных систем, баз данных, архитектур компьютерных приложений, и других систем (далее, моделей). С её помощью можно выделить ключевые сущности, присутствующие в модели, и обозначить отношения, которые могут устанавливаться между этими сущностями [4].

4.2 Логическое проектирование

Логическое проектирование базы данных системы проведено с помощью CASE - средства Sybase Powerdesigner 15.3.

На основе анализа предметной области выделены сущности:

					ПГУ 1.090301.04.001 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

а) сущность «чувствительный элемент» определяется атрибутами: идентификатор чувствительного элемента, название, описание;

б) сущность «датчик» определяется атрибутами: идентификатор датчика, название, описание;

в) сущность «объект» определяется атрибутами: идентификатор объекта, название, адрес;

г) сущность «канал связи» определяется атрибутами: идентификатор канала связи, название, описание;

д) сущность «установка датчика» определяется атрибутами: идентификатор объекта, идентификатор датчика, идентификатор канала связи, дата и время, активен, описание, идентификатор измерительного канала;

е) сущность «установка элементов» определяется атрибутами: идентификатор чувствительного элемента, идентификатор датчика, дата и время, активен, описание;

ж) сущность «измерительный канал» определяется атрибутами: идентификатор измерительного канала, название, идентификатор единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max;

и) сущность «единицы измерений» определяется атрибутами: идентификатор единицы измерения, название;

к) сущность «математический канал» определяется атрибутами: идентификатор математического канала, название, функция, идентификатор единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max;

л) сущность «пользователь» определяется атрибутами: идентификатор пользователя, фамилия, имя, отчество, дата рождения, идентификатор должности;

м) сущность «должность» определяется атрибутами: идентификатор должности, название;

н) сущность «действия пользователя» определяется атрибутами: дата и время, идентификатор пользователя, описание;

о) сущность «действия системы» определяется атрибутами: дата и время, идентификатор вида сообщения, сообщение;

п) сущность «вид сообщения» определяется атрибутами: идентификатор вида сообщения, название;

р) сущность «состояние» определяется атрибутами: дата и время, измерительный канал 1, измерительный канал 2, измерительный канал 3, измерительный канал 4, измерительный канал 5, измерительный канал 6, измерительный канал 7, измерительный канал 8, измерительный канал 9, измерительный канал 10, измерительный канал 11, измерительный канал 12, измерительный канал 13, измерительный канал 14, измерительный канал 15, измерительный канал 16, математический канал 1, математический канал 2, математический канал 3, математический канал 4, математический канал 5, математический канал 6.

Для того чтобы однозначно идентифицировать каждый экземпляр сущности - выделим первичные ключи:

а) сущность «чувствительный элемент» - первичный ключ «идентификатор чувствительного элемента»;

б) сущность «датчик» - первичный ключ «идентификатор датчика»;

в) сущность «объект» - первичный ключ «идентификатор объекта»;

г) сущность «канал связи» - идентификатор канала связи;

д) сущность «установка датчика» - составной первичный ключ, состоящий из: «идентификатор объекта» и «идентификатор датчика»;

е) сущность «установка элементов» - составной первичный ключ, состоящий из: «идентификатор чувствительного элемента» и «идентификатор датчика»;

ж) сущность «измерительный канал» - первичный ключ «идентификатор измерительного канала»;

и) сущность «единицы измерений» - первичный ключ «идентификатор единицы измерения»;

к) сущность «математический канал» - первичный ключ «идентификатор математического канала»;

л) сущность «пользователь» - первичный ключ «идентификатор пользователя»;

м) сущность «должность» - первичный ключ «идентификатор должности»;

н) сущность «действия пользователя» - первичный ключ «дата и время»;

о) сущность «действия системы» - первичный ключ «дата и время»;

п) сущность «вид сообщения» - первичный ключ «идентификатор вида сообщения»;

р) сущность «состояние» - первичный ключ «дата и время».

ER-диаграмма логической структуры базы данных была спроектирована с помощью CASE - средства Sybase Powerdesigner 15.3 и представлена на рисунке Б.1.

4.3 Физическое проектирование

На этапе логического проектирования осуществляется разбиение БД на таблицы. При этом каждой сущности предметной области ставится в соответствие таблица, атрибутам объекта соответствуют атрибуты таблицы, а идентификатору объекта соответствует ключ таблицы.

Типы данных атрибутов сущностей приведены в таблицах 4.1-4.15.

Таблица 4.1 - Типы данных атрибутов сущности «должность»

Атрибут	Тип данных
Идентификатор должности	Serial
Название	Text

Таблица 4.2 - Типы данных атрибутов сущности «вид сообщения»

Атрибут	Тип данных
Идентификатор вид сообщения	Serial
Название	Text

Таблица 4.3 - Типы данных атрибутов сущности «единицы измерения»

Атрибут	Тип данных
Идентификатор единицы измерения	Serial
Название	Text

Таблица 4.4 - Типы данных атрибутов сущности «чувствительный элемент»

Атрибут	Тип данных
Идентификатор чувствительного элемента	Serial
Название	Text
Описание	Text

Таблица 4.5 - Типы данных атрибутов сущности «пользователь»

Атрибут	Тип данных
Идентификатор пользователя	Serial
Фамилия	Text
Имя	Text
Отчество	Text
Дата рождения	Date
Идентификатор должности	Integer

Таблица 4.6 - Типы данных атрибутов сущности «датчик»

Атрибут	Тип данных
Идентификатор датчика	Serial
Название	Text
Описание	Text

Таблица 4.7 - Типы данных атрибутов сущности «объект»

Атрибут	Тип данных
Идентификатор объекта	Serial
Название	Text
Адрес	Text

Таблица 4.8 - Типы данных атрибутов сущности «действия пользователя»

Атрибут	Тип данных
Дата и время	Date & Time
Идентификатор пользователя	Integer
Описание	Text

Таблица 4.9 - Типы данных атрибутов сущности «действия системы»

Атрибут	Тип данных
Дата и время	Date & Time
Идентификатор вида сообщения	Integer
Сообщение	Text

Таблица 4.10 - Типы данных атрибутов сущности «канал связи»

Атрибут	Тип данных
Идентификатор канала связи	Serial
Название	Text
Описание	Text

Таблица 4.11 - Типы данных атрибутов сущности «измерительный канал»

Атрибут	Тип данных
Идентификатор измерительного канала	Serial
Название	Text
Идентификатор единицы измерения	Integer
ПДК max	Float
ПДК min	Float
КЗ max	Float
КЗ min	Float

Таблица 4.12 - Типы данных атрибутов сущности «математический канал»

Атрибут	Тип данных
Идентификатор математического канала	Serial
Название	Text
Функция	Text
Идентификатор единицы измерения	Integer
ПДК min	Float
ПДК max	Float
КЗ min	Float
КЗ max	Float

Таблица 4.13 - Типы данных атрибутов сущности «установка элементов»

Атрибут	Тип данных
Идентификатор чувствительного элемента	Integer
Идентификатор датчика	Integer
Активен	Boolean
Описание	Text

Таблица 4.14 - Типы данных атрибутов сущности «установка датчика»

Атрибут	Тип данных
Идентификатор объекта	Integer
Идентификатор датчика	Integer
Идентификатор канала связи	Integer
Активен	Boolean
Описание	Text
Идентификатор измерительного канала	Integer

Таблица 4.15 - Типы данных атрибутов сущности «состояние»

Атрибут	Тип данных
Дата и время	Date & Time
Измерительный канал 1	Float
Измерительный канал 2	Float
Измерительный канал 3	Float
Измерительный канал 4	Float
Измерительный канал 5	Float
Измерительный канал 6	Float
Измерительный канал 7	Float
Измерительный канал 8	Float
Измерительный канал 9	Float
Измерительный канал 10	Float
Измерительный канал 11	Float
Измерительный канал 12	Float
Измерительный канал 13	Float
Измерительный канал 14	Float
Измерительный канал 15	Float
Измерительный канал 16	Float
Математический канал 1	Float
Математический канал 2	Float
Математический канал 3	Float
Математический канал 4	Float
Математический канал 5	Float
Математический канал 6	Float

ER-диаграмма физической структуры базы данных была спроектирована с помощью CASE - средства Sybase Powerdesigner 15.3 и представлена на рисунке Б.2.

4.4 Разработка структуры связей

Между объектами предметной области существуют связи, которые должны быть отражены в виде связей между объектами инфологической модели. Графически связь обозначается линией, соединяющей связываемые объекты. В каждом направлении связи можно выделить главный объект, от которого идет связь, и подчиненный.

Различают идентифицирующую связь и неидентифицирующую связь. При установлении неидентифицирующей связи дочерняя сущность остается

независимой. Экземпляр сущности родителя может существовать безотносительно к какому-либо экземпляру дочерней сущности.

Сущности «Чувствительный элемент» и «Установка элементов» связаны через внешний ключ по полю «идентификатор чувствительного элемента». Связь идентифицирующая. Один чувствительный элемент может устанавливаться в разные датчики, но в один датчик может устанавливаться только один чувствительные элементы. Поэтому связь между сущностями - «один ко многим».

Сущности «Установка элементов» и «Датчик» связаны через внешний ключ по полю «идентификатор датчика». Связь идентифицирующая. Один чувствительный элемент может устанавливаться только в один датчик, но в один датчик могут устанавливаться разные чувствительные элементы. Поэтому связь между сущностями - «один ко многим».

Сущности «Датчик» и «Установка датчика» связаны через внешний ключ по полю «идентификатор датчика». Связь идентифицирующая. В каждой установке датчика на объект устанавливается только один датчик, но датчики могут устанавливаться на разные объекты, следовательно, связь между сущностями - «один ко многим».

Сущности «Объект» и «Установка датчика» связаны через внешний ключ по полю «идентификатор объекта». Связь идентифицирующая. При установке датчик устанавливается на один объект, но на один объект могут устанавливаться несколько датчиков, следовательно, связь между сущностями - «один ко многим».

Сущности «Канал связи» и «Установка датчика» связаны через внешний ключ по полю «идентификатор канала связи». Связь неидентифицирующая. При использовании датчика может устанавливаться только один канал связи, но один канал связи может использоваться в разных датчиках. Поэтому связь между сущностями - «один ко многим».

Сущности «Измерительный канал» и «Установка датчика» связаны через внешний ключ по полю «идентификатор измерительный канал». Связь

					ПГУ 1.090301.04.001 ПЗ	Лист
						38
Изм.	Лист	№ докум.	Подпись	Дата		

неидентифицирующая. На один датчик выделяется один канал, но один канал может использоваться несколькими датчиками. Поэтому связь между сущностями - «один ко многим».

Сущности «Измерительный канал» и «Единицы измерения» связаны через внешний ключ по полю «идентификатор единицы измерения». Связь неидентифицирующая. На каждом канале устанавливаются единицы измерения, но одни и те же единицы измерения могут использоваться на разных каналах. Поэтому связь между сущностями - «один ко многим».

Сущности «Математический канал» и «Единицы измерения» связаны через внешний ключ по полю «идентификатор единицы измерения». Связь неидентифицирующая. На каждом математическом канале устанавливается единица измерения, но одна и та же единица измерения может использоваться на разных каналах. Поэтому связь между сущностями - «один ко многим».

Сущности «Должность» и «Пользователь» связаны через внешний ключ по полю «идентификатор должность». Связь неидентифицирующая. Пользователь может занимать только одну должность, но одна должность может быть у нескольких пользователей. Поэтому связь между сущностями - «один ко многим».

Сущности «Пользователь» и «Действия пользователя» связаны через внешний ключ по полю «идентификатор пользователя». Связь неидентифицирующая. Пользователь может совершать одновременно только одно действие, но одно и то же действие могут совершать различные пользователи. Поэтому связь между сущностями - «один ко многим».

Сущности «Действия системы» и «Вид сообщения» связаны через внешний ключ по полю «идентификатор вида сообщения». Связь неидентифицирующая. У сообщения может быть только один вид сообщения, но один вид сообщения может быть у нескольких сообщений. Поэтому связь между сущностями - «один ко многим».

Логическая структура базы данных (ER – диаграмма) представлена на рисунке Б.1.

					ПГУ 1.090301.04.001 ПЗ	Лист
						39
Изм.	Лист	№ докум.	Подпись	Дата		

На рисунке Б.2 представлена ER-диаграмма физического уровня.

4.5 Нормализация базы данных

Процесс нормализации выполняется путем анализа отношений с учетом их первичного ключа и существующих функциональных зависимостей. Он включает ряд правил, которые используются для проверки отношений, чтобы вся база данных могла быть нормализована до желаемой степени нормализации. Если некоторое требование не удовлетворяется, то должна быть проведена декомпозиция отношения на отношения, каждое из которых удовлетворяет всем требованиям нормализации [5].

При работе с реляционными базами данных обязательным является удовлетворение только требованиям первой нормальной формы (1НФ). Все остальные формы нормализации могут использоваться по желанию проектировщиков. Однако рекомендуется выполнять нормализацию как минимум до 3НФ, чтобы получить хорошее качество отношений (приемлемую избыточность отношений).

Атрибут, значение которого атомарно (неделимо), называется простым атрибутом. Сложный атрибут получается соединением нескольких атомарных атрибутов, которые могут быть определены на одном или разных доменах (его также называют вектор или агрегат данных) [5].

Ненормализованное отношение характеризуется таблицей, содержащей в одном или нескольких столбцах, повторяющиеся группы данных - сложные атрибуты (массив или вектор значений).

Отношение, в котором на пересечении каждой строки и каждого столбца содержится атомарное (или единственное) значение, находится в 1НФ. При этом необходимо, чтобы отношение имело первичный ключ. Для преобразования ненормализованной таблицы в первую нормальную форму, следует найти в исходной таблице и устранить все повторяющиеся группы данных, используя декомпозицию этой таблицы [5].

					ПГУ 1.090301.04.001 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Вторая нормальная форма применяется к отношениям с составными ключами, т.е. К таким отношениям, первичный ключ которых состоит из двух или больше атрибутов. Отношение с первичным ключом на основе единственного атрибута всегда находится в 2НФ.

Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов [5].

Отношение находится в 3НФ, если оно представлено в 2НФ и не имеет не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

Нормализация 2НФ-отношения с образованием 3НФ-отношения осуществляется путем устранения транзитивных зависимостей - транзитивно-зависимые атрибуты удаляются из отношения и помещаются в новое отношение вместе с их детерминантом [6].

Все атрибуты всех сущностей в данной проектируемой системе атомарны и находятся в 1 НФ. Рассмотрим для подтверждения данного факта сущность «канал связи». Атрибуты данной сущности «ид канал связи», «название», «описание» являются простыми и неделимыми.

2НФ требует, чтобы неключевые атрибуты отношений зависели от первичного ключа в целом, но не от его части. 2НФ применяется к отношениям с составными ключами, т.е. К таким отношениям, первичный ключ которых состоит из двух и более атрибутов.

Рассмотрим отношение «установка датчика», первичный ключ которого состоит из атрибутов «идентификатор объект» и «идентификатор датчик». Неключевые атрибуты данного отношения «идентификатор канал связи», «описание», «идентификатор канал» зависят от данного первичного ключа в целом, а не от его части, поэтому можно сделать вывод, что данное отношение находится во 2НФ.

Ни в одном из отношений не существует транзитивных зависимостей, т.е. Не ключевые атрибуты не зависят друг от друга, отношения находятся в 3НФ.

Примером служит сущность «пользователь». Неключевой атрибут «фамилия» не зависит от неключевых атрибутов «имя», «отчество», «дата рождения», «идентификатор должность». Поэтому можно сделать вывод, что данное отношение находится во 3НФ.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

5 Анализ целесообразности использования средств, языков и технологий разработки

В соответствии с требованиями ТЗ программное обеспечение АРМ оператора системы контроля газового состава должно быть спроектировано с помощью CASE средства Sybase Powerdesigner 15 и реализовано в среде Microsoft Visual Studio 2008 и выше, на языке программирования С#, технология .NET Framework 3.5, реализация базы данных с помощью СУБД Microsoft SQL Server 2008.

Powerdesigner - ведущий инструмент в области моделирования и управления метаданными, предназначенный для создания моделей данных, информационной архитектуры и архитектуры предприятия. Powerdesigner дает компании силу анализа взаимосвязей (технология impact analysis), возможность управления изменениями на этапе проектирования и технику управления метаданными.

Powerdesigner поддерживает моделирование данных: концептуальная, логическая и физическая модели данных, и расширение для проектирования хранилищ данных, основанное на IE (Information Engineering) методологии или нотации IDEF 1/x [7].

Powerdesigner поддерживает моделирование приложений: все диаграммы UML и управление реализацией персистентных объектов за счет расширенной возможности описания OR-отображений (Object/Relational mapping). Техники моделирования с использованием XML и связь таких моделей с UML и моделями данных.

Модели powerdesigner полностью интегрированы: используя уникальную технологию соединения и синхронизации (Link and Synch), powerdesigner интегрирует метаданные между всеми типами моделей.

Открытая поддержка всех основных платформ: более 60 СУБД, ведущие платформы для разработки приложений, такие как Java J2EE,

					ПГУ 1.090301.04.001 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

Microsoft .NET, Web Services и powerbuilder, а также языки исполнения процессов, такие как ebxml и BPEL4WS [7].

Методологическую основу powerdesigner составляет технология IDEF1X (моделирование данных для реляционных СУБД).

Несмотря на обилие языков программирования, в России в последнее время очень популярен C# - объектно-ориентированный язык программирования, разработанный в 1998 – 2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств, вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников – языков C++, Java, Delphi, Модула и Smalltalk – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов. При создании языка в основу дизайна легла легкость использования, доминирующая над мощностью языка и скоростью выполнения – память освобождается автоматически, отбирая при этом процессорное время. Также реализована безопасность работы с типами, а это, по мнению многих, является одним важнейшим фактором, позволяющим избежать ошибок. Работа с атрибутами позволяет использовать COM, COM+ и DLL в C# приложениях, работающих в среде .NET.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

Для хранения данных решено использовать реляционную базу данных Microsoft SQL Server 2008, которая является идеальным решением для клиент серверного приложения

Microsoft® SQL Server™ – это комплексная платформа баз данных, обеспечивающая управление данными в масштабе предприятия и оснащенная интегрированными средствами бизнес-аналитики (Business Intelligence, BI). Ядро СУБД SQL Server обеспечивает более безопасное и надежное хранение данных в реляционном формате и в формате XML. Благодаря поддержке как реляционного, так и XML-формата, SQL Server обеспечивает гибкость, необходимую для работы. Помимо поддержки реляционных баз данных и данных в формате XML, SQL Server 2005 также сочетает лучшее в таких областях, как анализ, подготовка отчетов, интеграция данных и рассылка уведомлений. Это позволяет создавать и развертывать экономически эффективные решения бизнес-аналитики, которые позволят сотрудникам компании доставлять аналитическую информацию в любое место предприятия при помощи карт ключевых показателей эффективности, панелей управления, веб-служб и мобильных устройств.

Выгодным преимуществом SQL Server 2008 является тесная интеграция с Microsoft Visual Studio, Microsoft Office System и целым комплектом новых средств разработки, включая Business Intelligence Development Studio. Эта платформа также предоставляет возможность создания и управления высокопроизводительными приложениями баз данных с высоким уровнем доступности.

Таким образом среда разработки Microsoft Visual Studio C# в сочетании с Microsoft SQL Server 2008 обладает необходимыми инструментами для быстрой и удобной разработки данной системы, и позволяет в перспективе интегрировать разработанную систему в глобальную систему мониторинга технически сложных объектов.

6 РЕАЛИЗАЦИЯ

6.1 Общие сведения

Разработанная в ходе выполнения бакалаврской работы была разработана автоматизированная система, которая называется «Программное обеспечение АРМ оператора системы контроля газового состава». Главный проектный файл называется monitoring.exe. При работе программа использует базу данных, файл которой называется monitoring.mdf. Шаблоны документов obdat.doc, pokaz.doc, gyrnal.doc. Также используются dll компоненты Interop.VBIDE.dll, Microsoft.Office.Interop.Word.dll, Microsoft.Vbe.Interop.dll, office.dll.

Программа работает в операционных системах Windows XP, Windows 7.

6.2 Функциональное назначение

Разработанное программное обеспечение предназначено для мониторинга газового состава технически сложных объектов нефтегазового комплекса.

Система предоставляет возможность ввода, редактирования и просмотра содержимого базы данных, а также составления отчетов. Программа имитирует сигналы с датчиков (с помощью псевдослучайной последовательности), с перспективной заменой на прием сигналов с многоканального электронного регистратора, ведет контроль 16 каналов, формирует 6 математических каналов для расчета необходимых параметров из отслеживаемых. Хранит данные о примененных датчиках и чувствительных элементах. Хранит информацию об объектах, на которых установлены датчики, и виде примененных каналов. Позволяет отслеживать состояния каждого канала в виде графиков, цифровых показаний (таблиц), шкал, вертикальных столбиковых диаграмм (цикл отображения (временной интервал вывода каждой следующей

					ПГУ 1.090301.04.001 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

точки на экран) задаётся пользователем при настройке. Система обеспечивает установку пределов нормального режима работы, установку параметров аварийного срабатывания, сигнализацию в случае выхода за пределы нормального режима работы, протоколирование действий пользователя и сообщений системы, вывод архивной информации.

6.3 Входные данные

Входными данными системы являются:

- а) показания датчиков (генератор случайных чисел, в перспективе показания с регистратора);
- б) информация об объектах (название, адрес, координаты);
- в) информация о датчиках (название, описание);
- г) информация о чувствительных элементах (название, описание);
- д) информация о каналах связи (название, описание);
- е) информация о каналах (название, единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max, описание);
- ж) информация о математических каналах (название, функция для вычисления, единицы измерения, ПДК min, ПДК max, КЗ min, КЗ max, описание);
- з) информация о пользователях (фамилия, имя, отчество, дата рождения, должность).

6.4 Выходные данные

Выходными данными системы являются:

- а) отчет об объектах и датчиках;
- б) отчет о состоянии каналов;
- в) журнал действий пользователей;
- г) журнал сообщений системы.

6.5 Описание логической структуры

В базе данных храниться информация о датчиках, чувствительных элементах, объектах, каналах, каналах связи, пользователях, действиях пользователей и системы.

С помощью среды программирования баз данных SQL Server Management Studio 2008 была создана клиент-серверная база данных. Файл базы данных называется monitoring.mdf.

Были созданы таблицы: «action_system» - действия системы, «action_users» - действия пользователей, «dat» - датчики, «dol» - должности, «e_i» - единицы измерения, «el» - чувствительные элементы, «k» - настройки каналов, «ks» - каналы связи, «m_k» - математические каналы, «objects» - объекты, «rnd» - установки генератора случайных чисел, «status» - состояние каналов, «type_message» - виды сообщений, «users» - пользователи, «yst_dat» - установка датчиков, «yst_el» - установка чувствительных элементов.

Таблицы «action_system» и «users» соединены между собой внешним ключом id_users.

Таблицы «action_users» и «users» соединены между собой внешним ключом id_users.

Таблицы «users» и «dol» соединены между собой внешним ключом id_dol.

Таблицы «el» и «yst_el» соединены между собой внешним ключом id_el.

Таблицы «dat» и «yst_el» соединены между собой внешним ключом id_dat.

Таблицы «dat» и «yst_dat» соединены между собой внешним ключом id_dat.

Таблицы «objects» и «yst_dat» соединены между собой внешним ключом id_ob.

Таблицы «ks» и «yst_dat» соединены между собой внешним ключом id_ks.

Таблицы «k» и «yst_dat» соединены между собой внешним ключом id_k.

Таблицы «k» и «e_i» соединены между собой внешним ключом id_e_i.

Таблицы «m_k» и «e_i» соединены между собой внешним ключом id_e_i.

Программа разработана в среде Microsoft Visual Studio 2010.

Проект сохранен в отдельном каталоге monitoring на локальном диске C:\ под именем monitoring.sln. В указанном каталоге также сохранены модули всех созданных форм.

Были разработаны следующие формы:

- вход в систему;
- главная форма;
- объекты;
- датчики;
- чувствительные элементы;
- установка чувствительных элементов;
- установка датчиков;
- настройка каналов;
- настройка математических каналов;
- настройки генератора случайных чисел;
- пользователи;
- должности;
- единицы измерения;
- каналы связи;
- виды сообщений;
- архив;
- журнал действий пользователей;

- журнал действий системы;
- мониторинг одного канала;
- мониторинг двух каналов;
- мониторинг четырех каналов.

Форма «Вход в систему» появляется при запуске системы, авторизует пользователя по логину и паролю.

Форма «Главная форма» появляется после авторизации в системе, позволяет начать работу с системой, либо изменить данные.

Форма «Объекты» содержит данные об объектах, позволяет их изменять, добавлять и удалять.

Форма «Датчики» содержит данные о датчиках, позволяет их изменять, добавлять и удалять.

Форма «Чувствительные элементы» содержит данные о чувствительных элементах, позволяет их изменять, добавлять и удалять.

Форма «Установка чувствительных элементов» содержит данные об установке в датчики чувствительных элементов, позволяет их изменять, добавлять и удалять.

Форма «Установка датчиков» содержит данные об установке датчиков на объекты, назначении им каналов связи и измерительных каналов, позволяет их изменять, добавлять и удалять.

Форма «Настройка каналов» содержит данные о настройках каналов, позволяет их изменять, добавлять и удалять.

Форма «Настройка математических каналов» содержит данные о настройках математических каналов, позволяет их изменять, добавлять и удалять.

Форма «Настройки генератора случайных чисел» содержит данные о настройках генератора случайных чисел, позволяет их изменять, добавлять и удалять.

Форма «Пользователи» содержит данные о пользователях, позволяет их изменять, добавлять и удалять.

Форма «Должности» содержит данные о должностях, позволяет их изменять, добавлять и удалять.

Форма «Единицы измерения» содержит данные о единицах измерения, позволяет их изменять, добавлять и удалять.

Форма «Каналы связи» содержит данные о каналах связи, позволяет их изменять, добавлять и удалять.

Форма «Виды сообщений» содержит данные о видах сообщений системы, позволяет их изменять, добавлять и удалять.

Форма «Архив» выводит архивную информацию о значениях измерительных каналов за выбранный период времени.

Форма «Журнал действий пользователей» выводит информацию о действиях пользователей за выбранный период времени.

Форма «Журнал действий системы» выводит информацию о действиях системы за выбранный период времени.

Форма «Мониторинг одного канала» позволяет отследить значение одного измерительного канала в виде графика, таблицы, шкалы, столбиковой диаграммы.

Форма «Мониторинг двух каналов» позволяет отследить значение двух измерительных каналов в виде графиков, таблиц, шкал, столбиковых диаграмм.

Форма «Мониторинг четырех каналов» позволяет отследить значение четырех измерительных каналов в виде графиков, таблиц, шкал, столбиковых диаграмм.

В ходе выполнения бакалаврской работы были разработаны следующие отчеты:

- а) отчет об объектах и датчиках;
- б) отчет о состоянии каналов;
- в) журнал действий пользователей;

г) журнал сообщений системы.

Текст модулей разработанной программы приведен в приложении Г.

Диаграмма взаимодействия программ представлена на рисунке Д.1. Модули, процедуры и функции данной программы перечислены в таблице 6.1.

					ПГУ 1.090301.04.001 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 6.1 – Структура программы

Модули	Процедуры и функции	Назначение
1	2	3
Initform (Вход в систему)	Private void button1_Click (object sender, EventArgs e)	Авторизация пользователя по логину и паролю
Main (Главная форма)	Private void чувствительные Элементы toolStrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Чувствительные элементы»
	Private void датчики toolStrip menuItem_Click (object sender, EventArgs e)	Вызов формы «Датчики»
	Private void объекты toolStrip menuItem_Click (object sender, EventArgs e)	Вызов формы «Объекты»
	Private void каналы связи tool- stripmenuItem_Click (object sender, EventArgs e)	Вызов формы «Каналы связи»
	Private void единицы Измерения toolStrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Единицы измерения»
	Private void должности Too Strip menuItem_Click (object sender, EventArgs e)	Вызов формы «Должности»
	Private void виды сообщений toolStripmenuItem_Click (object sender, EventArgs e)	Вызов формы «Виды сообщений»
	Private void каналы Tool stripmenuItem_Click (object sender, EventArgs e)	Вызов формы «Настройка каналов»
	Private void матканалы Tool stripmenuItem_Click (object sender, EventArgs e)	Вызов формы «Настройка математических каналов»
	Private void действия Пользователей toolStrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Журнал действий пользователей»

Продолжение таблицы 6.1

1	2	3
	Private void действиясистемы toolstripmenuitem_Click (object sender, EventArgs e)	Вызов формы «Журнал действий системы»
	Private void пользователи toolstripmenuitem_Click (object sender, EventArgs e)	Вызов формы «Пользователи»
	Private void гсч toolstrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Настройка генератора случайных чисел»
	Private void установка Сенсоров toolstrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Установка сенсоров»
	Private void установка Датчиков toolstrip menu- item_Click (object sender, EventArgs e)	Вызов формы «Установка датчиков»
	Private void main_Load (object sender, EventArgs e)	Отображение текущих показаний каждого канала
	Private void k1toolstrip menuitem_Click (object sender, EventArgs e)	Вызов формы «Мониторинг 1»
	Private void k2toolstrip menuitem_Click (object sender, EventArgs e)	Вызов формы «Мониторинг 2»
	Private void k4toolstrip menuitem_Click (object sender, EventArgs e)	Вызов формы «Мониторинг 4»
Об (Объекты)	Private void ob_Load(object sender, EventArgs e)	Отображение списка объектов
	Private void datagridview1_ cellclick (object sender, Data gridviewcelleventargs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи

Продолжение таблицы 6.1

1	2	3
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Dat (Датчики)	Private void dat_Load(object sender, EventArgs e)	Отображение списка датчиков
	Private void datagridview1_cellclick (object sender, datagridviewcelleventargs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
El (Чувствительные элементы)	Private void el_Load(object sender, EventArgs e)	Отображение списка чувствительных элементов
	Private void datagridview1_cellclick (object sender, datagridviewcelleventargs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Yst_el (Установка чувствительных элементов)	Private void yst_el_Load (object sender, EventArgs e)	Отображение списка установленных чувствительных элементов в датчики
	Private void datagridview1_cellclick (object sender, datagridviewcelleventargs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи

Продолжение таблицы 6.1

1	2	3
	Private void checkbox1_checkedchanged(object sender, EventArgs e)	Установка флага активен/неактивен
Yst_dat (Установка датчиков)	Private void yst_dat_Load(object sender, EventArgs e)	Отображение списка установленных чувствительных элементов в датчики
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
	Private void checkbox1_checkedchanged (object sender, EventArgs e)	Установка флага активен/неактивен
	Private void kan_Click (object sender, EventArgs e)	Вызов формы «Настройка каналов»
К (Настройка каналов)	Private void k_Load (object sender, EventArgs e)	Отображение списка настроек измерительных каналов
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Мк (Настройка каналов)	Private void mk_Load(object sender, EventArgs e)	Отображение списка настроек м каналов

Продолжение таблицы 6.1

1	2	3
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
	Private void f_Click (object sender, EventArgs e)	Вызов формы «Ввод функции»
Rnd (Настройки генератора случайных чисел)	Private void rnd_Load(object sender, EventArgs e)	Отображение списка настроек измерительных каналов
	Private void timer1_Tick(object sender, EventArgs e)	Формирование случайных чисел для всех каналов по заданным установкам
	Private void k1n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 1
	Private void k2n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 2
	Private void k3n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 3
	Private void k4n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 4
	Private void k5n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 5
	Private void k6n_valuechanged (object sender, EventArgs e)	Установка начального значения генератора случайных чисел канала 6

Продолжение таблицы 6.1

1	2	3
	Private void k7n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 7
	Private void k8n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 8
	Private void k9n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 9
	Private void k10n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 10
	Private void k11n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 11
	Private void k12n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 12
	Private void k13n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 13
	Private void k14n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 14
	Private void k15n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 15
	Private void k16n_valuechanged (object sender, eventargs e)	Установка начального значение генератора случайных чисел канала 16
	Private void nk1min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 1
	Private void nk2min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 2

Продолжение таблицы 6.1

1	2	3
	Private void nk3min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 3
	Private void nk4min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 4
	Private void nk5min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 5
	Private void nk6min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 6
	Private void nk7min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 7
	Private void nk8min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 8
	Private void nk9min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 9
	Private void nk10min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 10
	Private void nk11min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 11
	Private void nk12min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 12
	Private void nk13min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 13
	Private void nk14min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 14

Продолжение таблицы 6.1

1	2	3
	Private void nk15min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 15
	Private void nk16min_valuechanged(object sender, eventargs e)	Установка минимального значение генератора случайных чисел канала 16
	Private void nk1max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 1
	Private void nk2max_valuechanged (object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 2
	Private void nk3max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 3
	Private void nk4max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 4
	Private void nk5max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 5
	Private void nk6max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 6
	Private void nk7max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 7
	Private void nk8max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 8
	Private void nk9max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 9
	Private void nk10max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 10

Продолжение таблицы 6.1

1	2	3
	Private void nk11max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 11
	Private void nk12max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 12
	Private void nk13max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 13
	Private void nk14max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 14
	Private void nk15max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 15
	Private void nk16max_valuechanged(object sender, eventargs e)	Установка максимального значение генератора случайных чисел канала 16
Users (Пользователи)	Private void users_Load(object sender, eventargs e)	Отображение списка пользователей
	Private void addbutton_Click (object sender, eventargs e)	Добавление новой записи
	Private void edbutton_Click (object sender, eventargs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, eventargs e)	Удаление выбранной записи
Dol (Должности)	Private void dol_Load(object sender, eventargs e)	Отображение списка должностей
	Private void datagridview1_cellclick (object sender, Data gridview-celleventargs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, eventargs e)	Добавление новой записи
	Private void edbutton_Click (object sender, eventargs e)	Изменение выбранной записи

Продолжение таблицы 6.1

1	2	3
Еi (Единицы измерения)	Private void ei_Load(object sender, EventArgs e)	Отображение списка единиц измерения
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Ks (Каналы связи)	Private void ks_Load (object sender, EventArgs e)	Отображение списка каналов связи
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Tm (Виды сообщений)	Private void tm_Load(object sender, EventArgs e)	Отображение списка видов сообщений
	Private void datagridview1_cellclick (object sender, DataGridViewCellEventArgs e)	Получение адреса в базе данных для изменения записи
	Private void addbutton_Click (object sender, EventArgs e)	Добавление новой записи
	Private void edbutton_Click (object sender, EventArgs e)	Изменение выбранной записи
	Private void delbutton_Click (object sender, EventArgs e)	Удаление выбранной записи
Archiv (Архив)	Private void archiv_Load(object sender, EventArgs e)	Отображение значений всех измерительных каналов за день

Продолжение таблицы 6.1

1	2	3
	Private void combobox1_selectedindexchanged(object sender, EventArgs e)	Выбор измерительного канала
	Private void datetimepicker1_valuechanged(object sender, EventArgs e)	Нижняя граница интервала времени
	Private void datetimepicker2_valuechanged(object sender, EventArgs e)	Верхняя граница интервала времени
	Private void view_Click (object sender, EventArgs e)	Отображение значений выбранного измерительного канала за выбранный интервал времени
	Private void checkbox1_checkedchanged(object sender, EventArgs e)	Флаг отображать все каналы или выбрать один
Actions_users (Журнал действий пользователей)	Private void archiv_Load(object sender, EventArgs e)	Отображение действий всех пользователей
	Private void combobox1_selectedindex Changed (object sender, EventArgs e)	Выбор пользователя
	Private void datetimepicker1_valuechanged(object sender, EventArgs e)	Нижняя граница интервала времени
	Private void datetimepicker2_valuechanged(object sender, EventArgs e)	Верхняя граница интервала времени
	Private void view_Click (object sender, EventArgs e)	Отображение действий выбранного пользователя за выбранный интервал времени
	Private void checkbox1_Checked Changed(object sender, EventArgs e)	Флаг отображать действия всех пользователей или выбрать один
Ac-tions_system (Журнал действий системы)	Private void archiv_Load(object sender, EventArgs e)	Отображение всех сообщений системы

Продолжение таблицы 6.1

1	2	3
	Private void combobox1_selectedindexchanged(object sender, EventArgs e)	Выбор вида сообщений системы
	Private void dateTimePicker1_valuechanged(object sender, EventArgs e)	Нижняя граница интервала времени
	Private void dateTimePicker2_valuechanged(object sender, EventArgs e)	Верхняя граница интервала времени
	Private void view_Click (object sender, EventArgs e)	Отображение сообщений выбранного типа за выбранный интервал времени
	Private void checkBox1_checkedchanged(object sender, EventArgs e)	Флаг отображать действия всех пользователей или выбрать один
Monitor1 (Мониторинг одного канала)	Private void checkBox1_checkedchanged(object sender, EventArgs e)	Выбор измерительного канала
	Private void gr_checkedchanged(object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void sd_checkedchanged(object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void tab_checkedchanged(object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void shkal_checkedchanged(object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void arhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void s1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с

Продолжение таблицы 6.1

1	2	3
	Private void s2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void s5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void s10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void s30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void m1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void m2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void m5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
Monitor2 (Мониторинг двух каналов)	Private void checkbox1_ checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void lkgr_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void lkzd_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void lktab_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void lksh- kal_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void lkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра значений выбранного канала
	Private void lks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void lks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с

Продолжение таблицы 6.1

1	2	3
	Private void lks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void lks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void lks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void lkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void lkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void lkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
	Private void checkbox22_ checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void pkgr_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void pkzd_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void pktab_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void pksh- kal_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void pkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void pks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void pks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void pks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с

Продолжение таблицы 6.1

1	2	3
	Private void pks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void pks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void pkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void pkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void pkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
Monitor4 (Мониторинг четырёх каналов)	Private void checkbox1_ checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void lkgr_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void lksd_Checked Changed (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void lktab_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void lkshkal_Checked Changed (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void lkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void lks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void lks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void lks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void lks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с

Продолжение таблицы 6.1

1	2	3
	Private void lks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void lkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void lkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void lkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
	Private void checkbox22_ checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void pkgr_ checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void pkd_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void pktab_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void pksh- kal_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void pkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void pks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void pks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void pks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void pks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void pks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с

Продолжение таблицы 6.1

1	2	3
	Private void pkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void pkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void pkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
	Private void checkbox1_ checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void lkgr_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void lkst_ checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void lktab_ Checked Changed (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void lkshkal_ Checked Changed (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void lkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void lks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void lks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void lks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void lks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void lks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void lkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м

Продолжение таблицы 6.1

1	2	3
	Private void lkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м
	Private void lkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м
	Private void checkbox22_checkedchanged (object sender, EventArgs e)	Выбор измерительного канала
	Private void pkgr_CheckedChanged(object sender, EventArgs e)	Вывод значений измерительного канала в виде графика
	Private void pksd_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде столбиковой диаграммы
	Private void pktab_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде таблицы
	Private void pkshkal_checkedchanged (object sender, EventArgs e)	Вывод значений измерительного канала в виде шкалы
	Private void pkarhiv_Click (object sender, EventArgs e)	Вызов формы «Архив» и просмотра на ней значений выбранного канала
	Private void pks1_Click (object sender, EventArgs e)	Установка цикла отображения = 1с
	Private void pks2_Click (object sender, EventArgs e)	Установка цикла отображения = 2с
	Private void pks5_Click (object sender, EventArgs e)	Установка цикла отображения = 5с
	Private void pks10_Click (object sender, EventArgs e)	Установка цикла отображения = 10с
	Private void pks30_Click (object sender, EventArgs e)	Установка цикла отображения = 30с
	Private void pkm1_Click (object sender, EventArgs e)	Установка цикла отображения = 1м
	Private void pkm2_Click (object sender, EventArgs e)	Установка цикла отображения = 2м

Продолжение таблицы 6.1

1	2	3
	Private void pkm5_Click (object sender, EventArgs e)	Установка цикла отображения = 5м

6.7 Описание запросов к системе

При отсутствии файла базы данных или ее повреждении, есть возможность пересоздать ее с помощью запросов. Пример запроса на создание таблицы «Датчики» приведен на рисунке 6.1.

```

dat.sql - ANKE...R\
USE [monitc]
GO
/***** Object: Table [dbo].[dat]
Script Date: 30/05/2017 08:58:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[dat] (
    [id_dat] [int] IDENTITY(1,1) NOT NULL,
    [dat] [nvarchar] (max) NOT NULL,
    [text] [nvarchar] (max) NOT NULL,
    CONSTRAINT [PK_dat] PRIMARY KEY CLUSTERED
(
    [id_dat] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
  
```

Рисунок 6.1 - Запрос на создание таблицы «Датчики»

Основным объектом хранения базы данных является таблица, поэтому все SQL-запросы — это операции над таблицами. В соответствии с этим, запросы делятся на:

- Запросы, оперирующие самими таблицами (создание и изменение таблиц);
- Запросы, оперирующие с отдельными записями (или строками таблиц) или наборами записей.

Запросы первого типа, в свою очередь, делятся на запросы, предназначенные для создания в базе данных новых таблиц, и на запросы, предназначенные для изменения уже существующих таблиц. Запросы второго типа оперируют со строками, и их можно разделить на запросы следующего вида:

- вставка новой строки;
- изменение значений полей строки или набора строк;
- удаление строки или набора строк.

Самый главный вид запроса — это запрос, возвращающий (пользователю) некоторый набор строк, с которым можно осуществить одну из трёх операций:

- просмотреть полученный набор;
- изменить все записи набора;
- удалить все записи набора.

Таким образом, использование SQL сводится, по сути, к формированию всевозможных выборок строк и совершению операций над всеми записями, входящими в набор

Логика работы системы основана на выполнении SQL-запросов к базе данных для получения необходимой информации. Впоследствии эта информация используется для расчетов, графического отображения состояния объектов и формирования отчетов. Все запросы к базе данных выполнены в виде хранимых процедур, для многократного выполнения.

Пример запроса добавления записи в таблицу «Датчики» представлен на рисунке 6.2.

Пример запроса изменения записи в таблице «Датчики» представлен на рисунке 6.3.

Пример удаления записи из таблицы «Датчики» представлен на рисунке 6.4.

Пример запроса получения данных о датчике по его идентификатору представлен на рисунке 6.5.

```

dat.sql - ANKE...R\
USE [monitoring]
GO
/***** Object: StoredProcedure [dbo].[dat_InsertQuery]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[dat_InsertQuery]
(
    @dat nvarchar(MAX),
    @text nvarchar(MAX)
)
AS
    SET NOCOUNT OFF;
INSERT INTO dat
VALUES (@dat, @text)
GO
/ ***/
GO

```

Рисунок 6.2 - Запрос добавления записи в таблицу «Датчики»

```

dat.sql - ANKE...R\
USE [monitoring]
GO
/***** Object: StoredProcedure [dbo].[dat_UpdateQuery]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[dat_UpdateQuery]
(
    @dat nvarchar(MAX),
    @text nvarchar(MAX),
    @Original_id_dat int
)
AS
    SET NOCOUNT OFF;
UPDATE dat
SET dat = @dat, text = @text
WHERE (id_dat = @Original_id_dat)
GO

```

Рисунок 6.3 - Запрос изменения записи в таблице «Датчики»

```

dat.sql - ANKE...R\
USE [monitoring]
GO
/***** Object: StoredProcedure [dbo].[dat_DeleteQuery]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[dat_DeleteQuery]
(
    @Original_id_dat int
)
AS
    SET NOCOUNT OFF;
DELETE FROM [dbo].[dat] WHERE ([id_dat] =
@Original_id_dat)
GO

```

Рисунок 6.4 - Запрос удаления записи из таблицы «Датчики»

```

dat.sql - ANKE...R\
USE [monitoring]
GO
/***** Object:  StoredProcedure [dbo].[dat_DataByID]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[dat_DataByID]
(
    @Param1 int
)
AS
    SET NOCOUNT ON;
SELECT      id_dat, dat, text
FROM        dat
WHERE       (id_dat = @Param1)
GO

```

Рисунок 6.5 – Запрос получения данных о датчике по его идентификатору

6.6 Используемые технические средства

Программное обеспечение разработано на ПК со следующими характеристиками:

- а) процессор Pentium(R) Dual-Core с тактовой частотой 2,7 гГц;
- б) оперативная память - 2 гбайт;
- в) свободное дисковое пространство - 500 мбайт;
- г) принтер.
- д)

6.7 Вызов и загрузка

Вызов программы осуществляется открытием EXE файл под названием monitoring.

7 Программа и методика испытаний

7.1 Объект и цель испытаний

Объектом испытания является программное обеспечение АРМ оператора системы контроля газового состава. Система реализует функции, оговоренные в техническом задании. Целью испытаний является проверка надежности и правильности выполнения всех функций системы.

7.2 Требования к программе

Требования к функциональным характеристикам программы описаны в техническом задании (пункт 2.3.1). Приложение должно обеспечивать выполнение всех перечисленных функций. Кроме того, программа должна гарантировать обработку ошибок пользователя (ввод неправильных символов, оставление незаполненными обязательных полей и т.п.). Также система должна выполнять контроль ссылочной целостности таблиц базы данных.

7.3 Методы испытаний

В процессе разработки программы была применена методология поэтапного тестирования, на каждом из этапов создания программы производилось её тестирование в пределах реализованных функциональных возможностей.

Тестирование программы в полном объеме предполагает её прогон по всем возможным вариантам выполнения.

Сохранение, выбор, изменение и удаление объектов системы осуществляется автоматически средствами встроенного языка и не требует проверки. Контроль ссылочной целостности также осуществляется средствами системы. Формирование псевдослучайной последовательности реализовано стандартной функцией языка программирования и не требует проверки.

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

Для тестирования выберем первые два канала. Введем ограничения нормального режима работы: измерительный канал № 1 ПДК $\min = 19$, ПДК $\max = 23$, критическое значение $\min = 15$, критическое значение $\max = 25$. Измерительный канал № 2 ПДК $\min = 0$, ПДК $\max = 2$, критическое значение $\min = 0$, критическое значение $\max = 2,5$. Генератор случайных чисел для первого канала: начальное значение $= 20$, $\min = 19$, $\max = 23$. Генератор случайных чисел для второго канала: начальное значение $= 1,5$, $\min = 0$, $\max = 2$. В ходе работы программы значения измерительных каналов должно колебаться в интервалах для первого канала 19-23, для второго 0-2,5, цвет поля зеленый. Далее увеличим $\max$ значение генератора случайных чисел для первого канала, установим равным 24. При генерации числа большего 23, цвет поля станет желтым и система выведет на экран сообщение о превышении предельно допустимой концентрации. Далее еще увеличим $\max$ значение генератора случайных чисел для первого канала, установим равным 26. При генерации числа большего 25, цвет поля станет красным и система выведет на экран сообщение о превышении критического значения.

7.4 Анализ результатов тестирования

Результаты тестирования программы представлены на рисунках 7.1 - 7.5.



Рисунок 7.1 - Нормальный режим работы

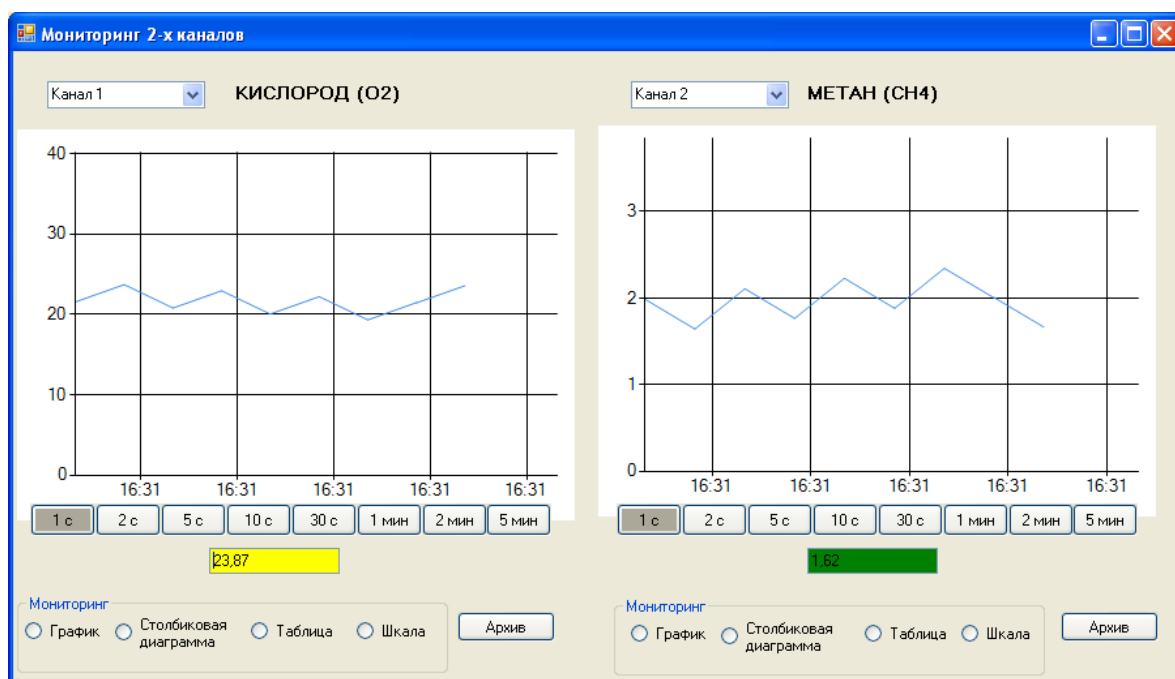


Рисунок 7.2 - Превышение ПДК max

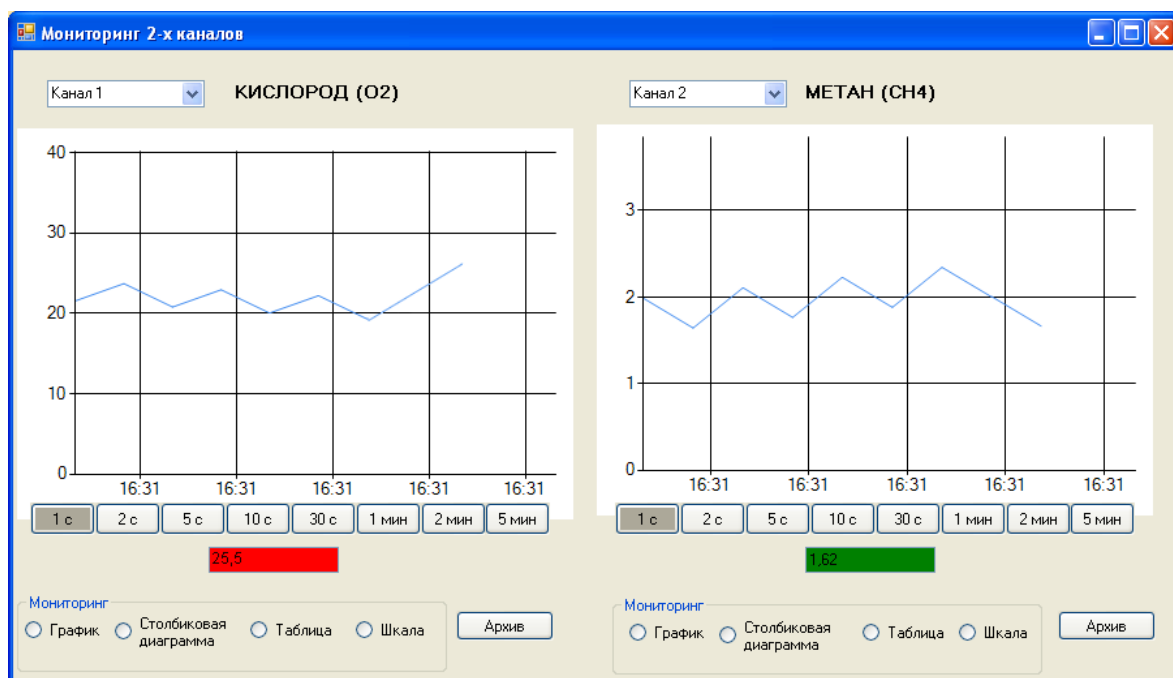


Рисунок 7.3 - Превышение критического значения max

Рисунок 7.4 - Предупреждение о превышении ПДК

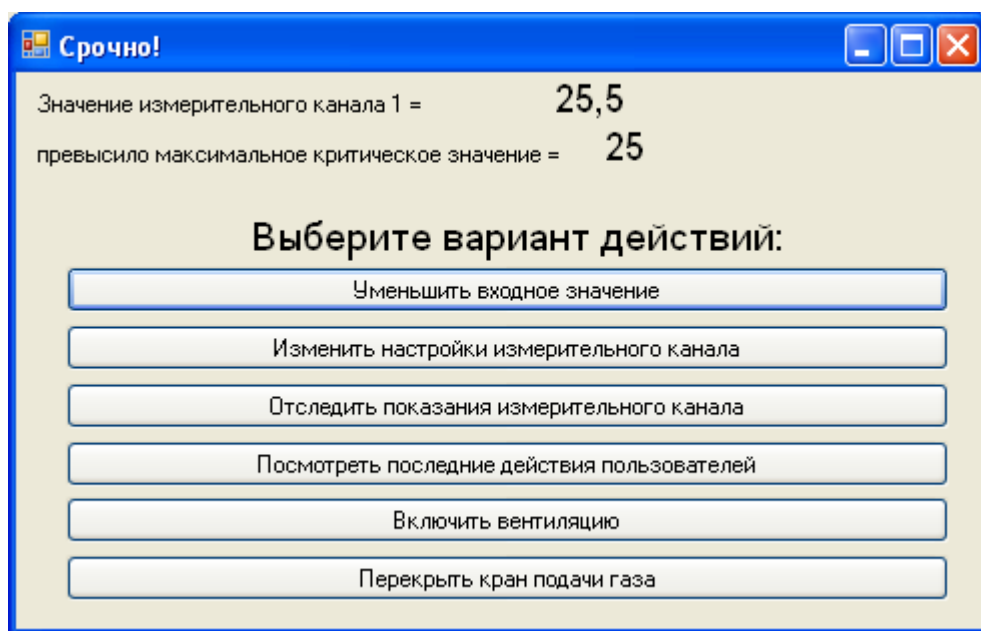


Рисунок 7.5 - Сообщение о превышении критических значений

Сравнив результаты работы программы с тестовым примером можно сделать вывод, что система функционирует корректно и обеспечивает все необходимые функции, заданные заказчиком.

8 Описание применения

При запуске программного обеспечения оператор системы контроля газового состава на форме «Авторизация» (рисунок 8.1) вводит логин и пароль.

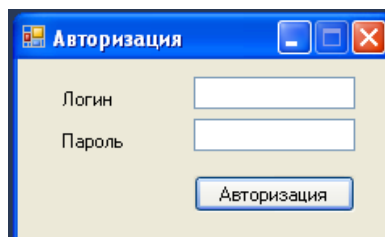


Рисунок 8.1 - Форма «Авторизация»

При некорректном вводе логина или пароля система возвращает информационное сообщение (рисунок 8.2).

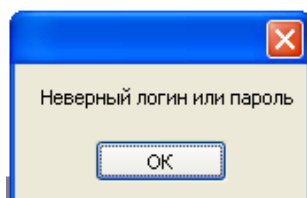


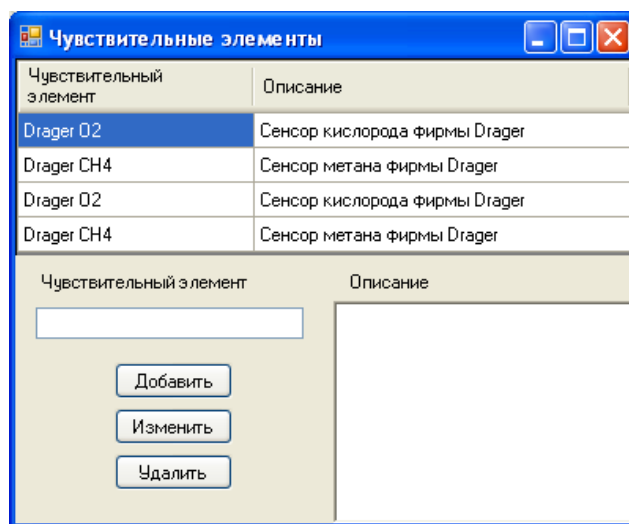
Рисунок 8.2 - Сообщение «Неверный логин или пароль»

Если вход в систему был произведен корректно, пользователь может приступить к заполнению первичных данных, выбрав соответствующий пункт меню на главной форме или наблюдать за состоянием измерительных каналов (рисунок 8.3).



Рисунок 8.3 - Главная форма

Выбрав пункт меню справочники оператор может вызвать формы: «Чувствительные элементы» (Рисунок 8.4), «Датчики» (Рисунок 8.5), «Каналы связи» (Рисунок 8.6), «Единицы измерения» (Рисунок 8.7), «Должности» (Рисунок 8.8), «Виды сообщений» (Рисунок 8.9) и ввести в систему первичную информацию.



Чувствительный элемент	Описание
Drager O2	Сенсор кислорода фирмы Drager
Drager CH4	Сенсор метана фирмы Drager
Drager O2	Сенсор кислорода фирмы Drager
Drager CH4	Сенсор метана фирмы Drager

Чувствительный элемент:

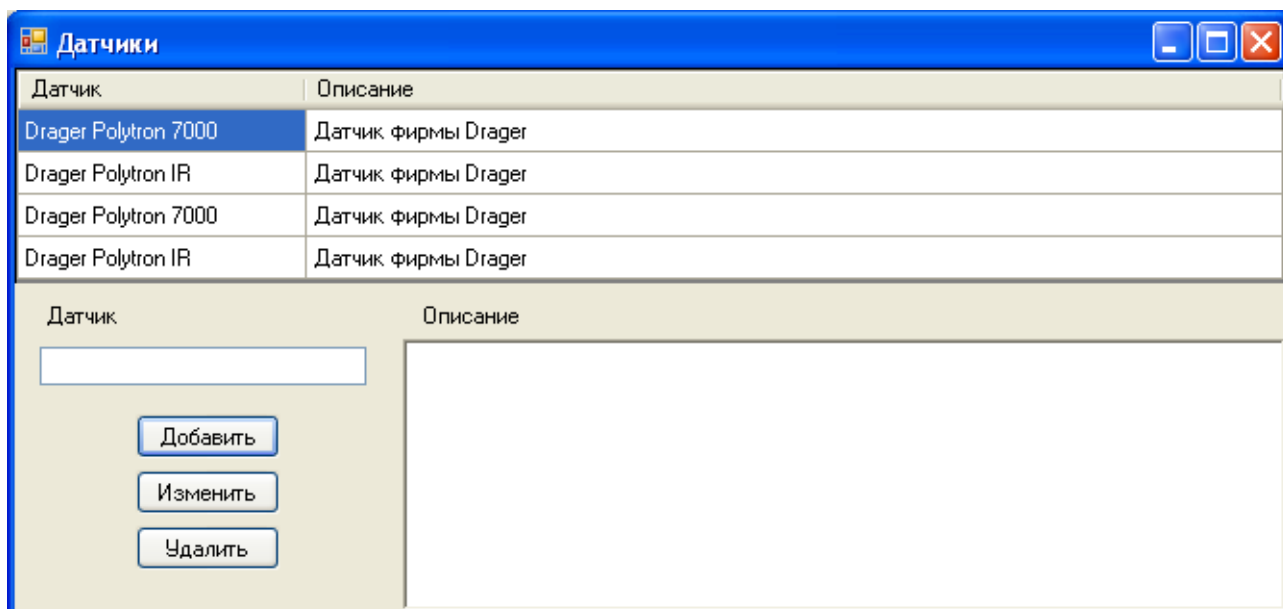
Описание:

Добавить

Изменить

Удалить

Рисунок 8.4 - Форма «Чувствительные элементы»



Датчик	Описание
Drager Polytron 7000	Датчик фирмы Drager
Drager Polytron IR	Датчик фирмы Drager
Drager Polytron 7000	Датчик фирмы Drager
Drager Polytron IR	Датчик фирмы Drager

Датчик:

Описание:

Добавить

Изменить

Удалить

Рисунок 8.5 - Форма «Датчики»

Канал связи	Описание
беспроводной Wi-Fi	Wi-Fi 208.11n
беспроводной GPS	GPS
ethernet 100	ethernet 100baseT

Канал связи:

Описание:

Добавить Изменить Удалить

Рисунок 8.6 - Форма «Каналы связи»

Единица измерения

M^3
%
dm^3
mm^3
ppt

Единица измерения:

Добавить Изменить Удалить

Рисунок 8.7 - Форма «Единицы измерения»

Должность

Техник 1 разряда
Начальник цеха
Техник 2 разряда

Должность:

Добавить Изменить Удалить

Рисунок 8.8 - Форма «Должности»

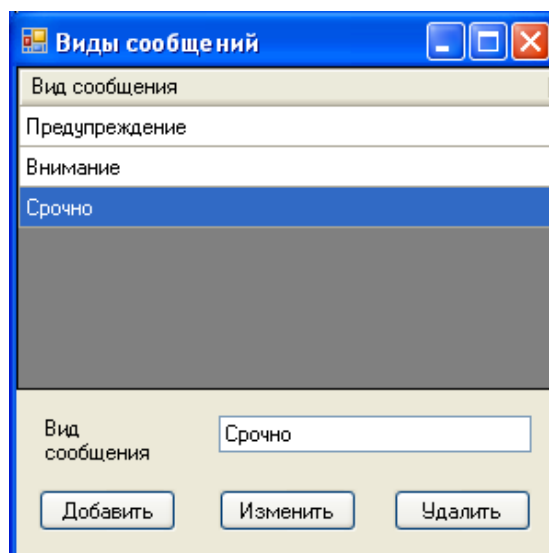


Рисунок 8.9 - Форма «Виды сообщений»

При необходимости оператор может изменить свои персональные данные (рисунок 8.10).

Рисунок 8.10 - Форма «Данные пользователя»

После того как первичная информация заполнена, необходимо занести информацию о том в какие датчики установлены чувствительные элементы (рисунок 8.11).

Датчик	Сенсор	Дата установки	Активен?	Описание
Drager Polytron 7000	Drager O2	12.04.2017	☒	
Drager Polytron IR	Drager CH4	12.04.2017	☒	szlhi

Датчик:

Сенсор:

☐ Активен

Добавить Изменить Удалить

Рисунок 8.11 - Форма «Установка сенсоров»

После установки чувствительных элементов, необходимо занести в систему данные об установке датчиков на объекты (рисунок 8.12).

Объект	Датчик	Дата установки	Канал связи	Измерительный канал	Активен?	Описание
Котельная	Drager Polytron 7000	16.04.2017	беспроводной ...	1	☒	
Завод Биосинтез	Drager Polytron IR	16.04.2017	беспроводной G...	4	☐	

Датчик:

Объект:

Канал связи:

Канал:

Настроить каналы

☐ Активен

Добавить Изменить Удалить

Рисунок 8.12 - Форма «Установка датчиков»

Нажав кнопку «Настроить каналы» оператор настраивает измерительные каналы (рисунок 8.13) или выбрав пункт меню настройки -> каналы.

№ изм. канала	Название изм. канала	Единица измерения	ПДК min	ПДК max	Критическое значение min	Критическое значение max	Описание
1	1	%	19	23	25	15	
2	2	%	0	2	0	2,5	
3	3	%	19	23	25	15	
4	4	%	0	2	0	2,5	

Канал № 1 Описание

Название канала:

Единицы измерения:

ПДК max:

ПДК min:

ПДК ЧС max:

ПДК ЧС min:

Рисунок 8.13 - Форма «Измерительные каналы»

Далее оператор настраивает математические каналы (рисунок 8.14), выбрав пункт меню настройки -> каналы. Для ввода функции (рисунок 8.15) для расчета параметра необходимо нажать кнопку «Изменить функцию».

№ мат_канала	Название мат_канала	Функция	Единица измерения	ПДК min	ПДК max	Критическое значение min	Критическое значение max	Описание
1	O2	$(k1 + k3) / 2$	%	23	19	25	15	
2	CH4	$(k2 + k4) / 2$	%	0	2	0	2,5	

Математический канал № Описание

Название канала:

Единицы измерения:

Функция:

ПДК max:

ПДК min:

ПДК ЧС max:

ПДК ЧС min:

Рисунок 8.14 - Форма «Математические каналы»

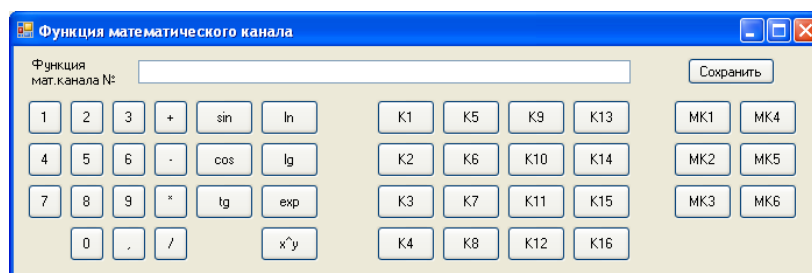


Рисунок 8.15 - Форма «Функция математического канала»

Далее оператор настраивает генератор случайных чисел (рисунок 8.16) для имитации сигналов с датчика, выбрав пункт меню настройки -> каналы.

Рисунок 8.16 - Форма «Настройка генератора случайных чисел»

Для мониторинга газового состава помещений технически сложных объектов можно отслеживать значения как одного (рисунок 8.17), так и двух (рисунок 8.18), измерительных каналов одновременно.

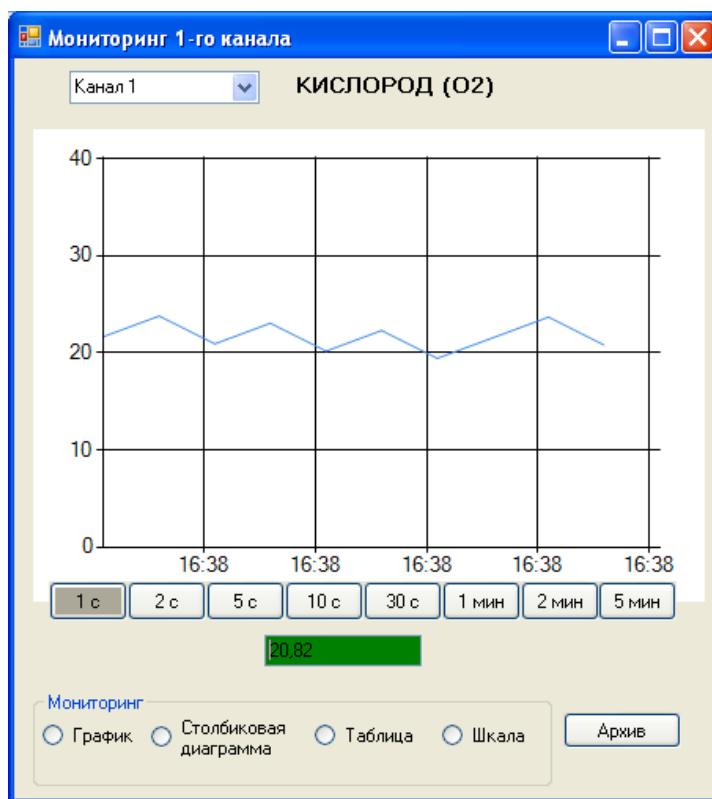


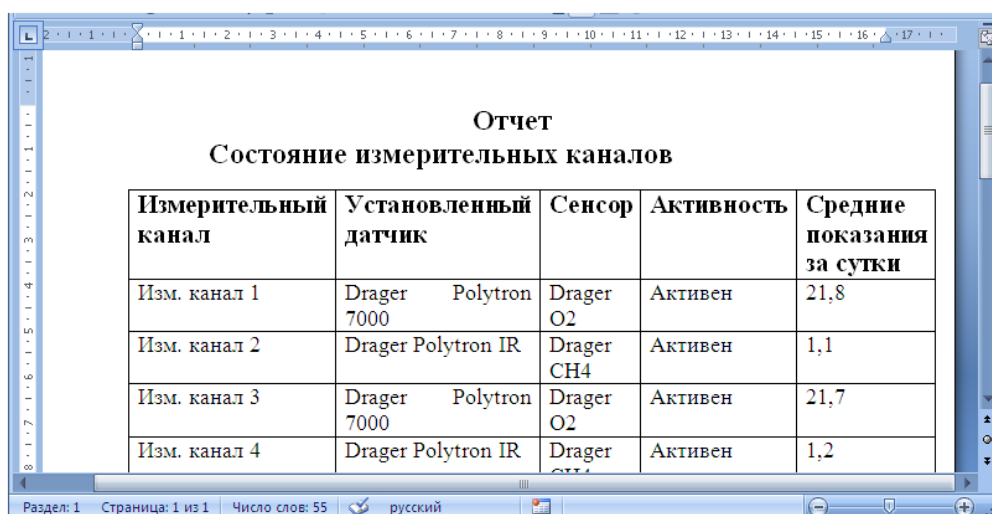
Рисунок 8.17 - Форма «Мониторинг 1-го канала»



Рисунок 8.18 - Форма «Мониторинг 2-х каналов»

Также при необходимости оператор может посмотреть сообщения системы и действия пользователей за определенный интервал времени.

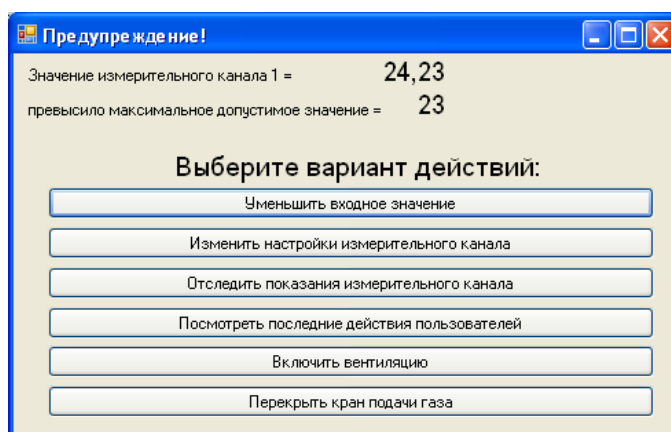
Выбрав на главной форме пункт меню «Журналы» оператор может посмотреть отчеты (рисунки 8.19)



Измерительный канал	Установленный датчик	Сенсор	Активность	Средние показания за сутки
Изм. канал 1	Drager Polytron 7000	Drager O2	Активен	21,8
Изм. канал 2	Drager Polytron IR	Drager CH4	Активен	1,1
Изм. канал 3	Drager Polytron 7000	Drager O2	Активен	21,7
Изм. канал 4	Drager Polytron IR	Drager CH4	Активен	1,2

Рисунок 8.19 - Отчет «Состояние измерительных каналов»

При превышении предельно допустимых значений или критических значений система отобразит на экране варианты решения (рисунки 8.20, 8.21).



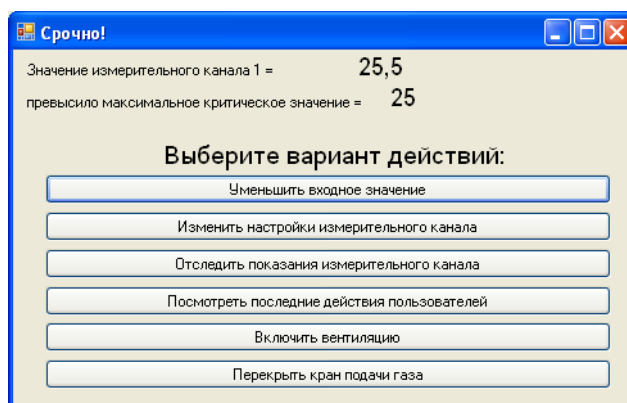
Предупреждение!

Значение измерительного канала 1 = 24,23
превысило максимальное допустимое значение = 23

Выберите вариант действий:

- Уменьшить входное значение
- Изменить настройки измерительного канала
- Отследить показания измерительного канала
- Посмотреть последние действия пользователей
- Включить вентиляцию
- Перекрыть кран подачи газа

Рисунок 8.20 - Предупреждение о превышении ПДК



Срочно!

Значение измерительного канала 1 = 25,5
превысило максимальное критическое значение = 25

Выберите вариант действий:

- Уменьшить входное значение
- Изменить настройки измерительного канала
- Отследить показания измерительного канала
- Посмотреть последние действия пользователей
- Включить вентиляцию
- Перекрыть кран подачи газа

Рисунок 8.21 - Сообщение о превышении критических значений

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы было разработано программное обеспечение АРМ оператора системы контроля газового состава для мониторинга состава газа в помещениях технически сложных объектов нефтегазового комплекса.

В ходе выполнения работы была изучена и проанализирована система контроля газового состава, собраны необходимые сведения и материалы, необходимые для разработки, выделены требования к системе.

Было проведено функциональное и инфологическое проектирование программного обеспечения с помощью CASE средства Sybase Powerdesigner, в ходе которого были разработаны диаграмма потоков данных и ее декомпозиция, позволяющая наглядно представить функциональные возможности программного обеспечения, и ER-диаграммы логического и физического.

Программное обеспечение было реализовано в среде программирования Microsoft Visual Studio 2010 на языке программирования С#. Тестирование программы показало, что она функционирует правильно, выполняет все функции, описанные в техническом задании на бакалаврскую работу.

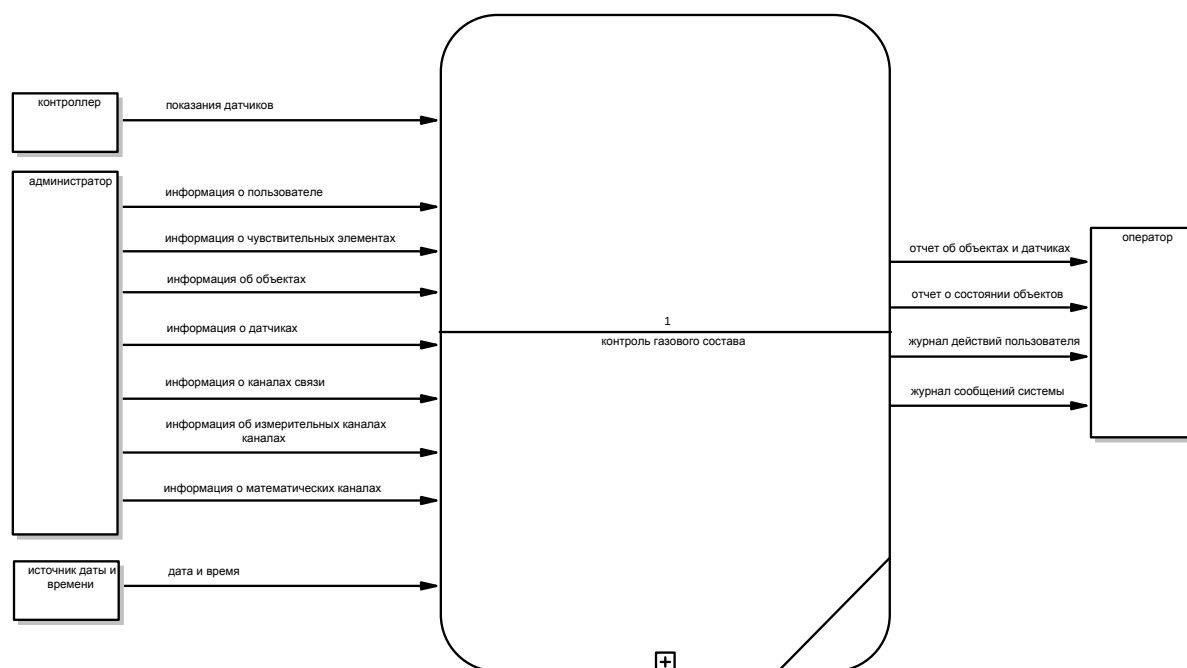
Таким образом, задание на бакалаврскую работу выполнено в полном объеме.

Список использованных источников

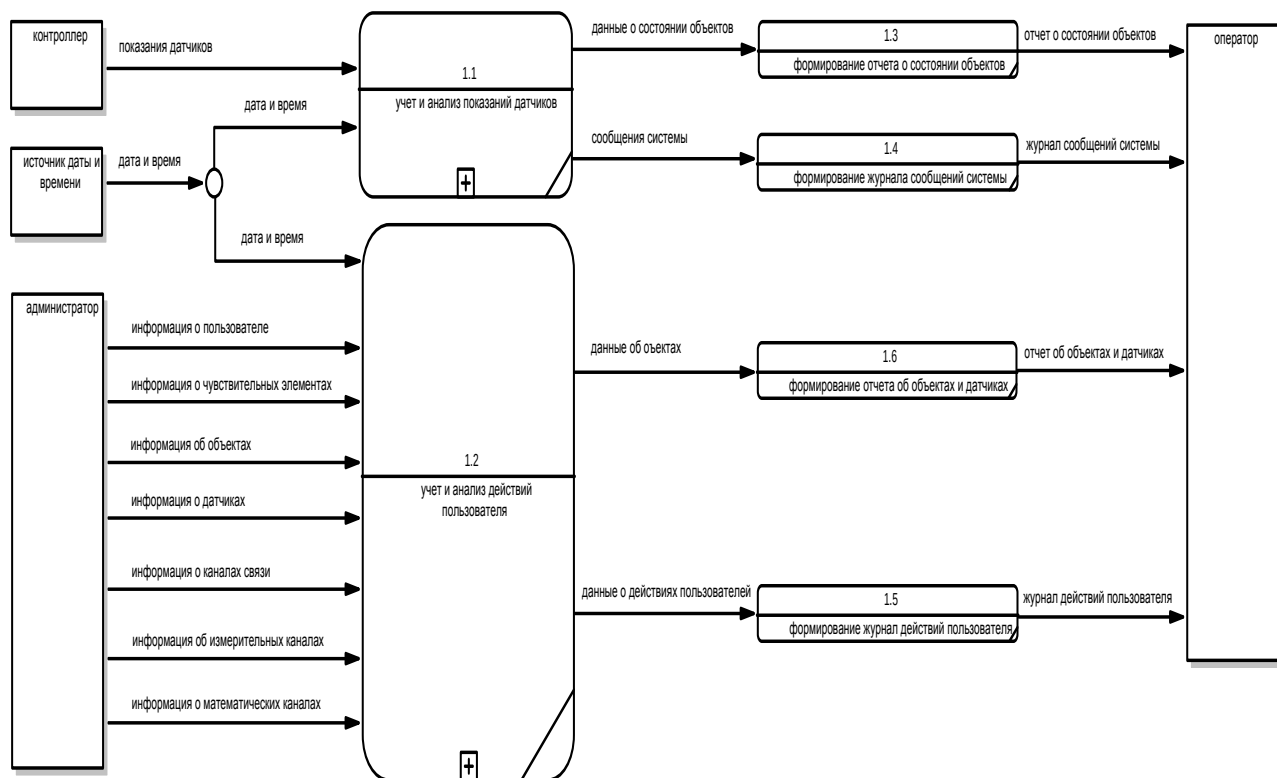
1. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2003. – 352с.
2. Постановление Минтруда России от 14 марта 1997 г. № 12 «О проведении аттестации рабочих мест по условиям труда». – Пермь: ДЕАН, 2004
3. Йессель В. Рекомендации по планированию и размещению систем газосигнализации. [Электронный ресурс]. – Электрон.дан. – URL: <http://drager.perm.ru/index.php?Page=files&item=1>
4. Интерфейс Ltd. [Электронный ресурс]. – Электрон.дан. – URL: <http://www.interface.ru>
5. Верников Г. Основы методологии IDEF1X. [Электронный ресурс]. – Электрон.дан. – URL: <http://www.interface.ru/fset.asp?Url=/ca/idef1x.htm>
6. Зайцев С. Л. Понятие отношения. [Электронный ресурс]. – Электрон.дан. – URL: <http://www.interface.ru/fset.asp?Url=/ca/ponatie.htm>
7. Нормальная форма. [Электронный ресурс]. – Электрон.дан. – URL: <http://ru.wikipedia.org>
8. 24. Закас Н. JavaScript. Оптимизация производительности. – Пер. с англ. – СПб.: Символ-Плюс, 2012. – 256 с.
9. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. –М.: Издательский дом "Вильямс", 2003. – 1440 с.
10. Основы проектирования реляционных баз данных / Туманов В.Е. – М.: Национальный Открытый Университет "ИНТУИТ", 2016 (Основы информационных технологий).
11. Официальный сайт powerdesigner. [Электронный ресурс]. – Электрон.дан. – URL: <http://www.sybase.ru/products/powerdesigner>

ПРИЛОЖЕНИЕ А
(Графическая часть)

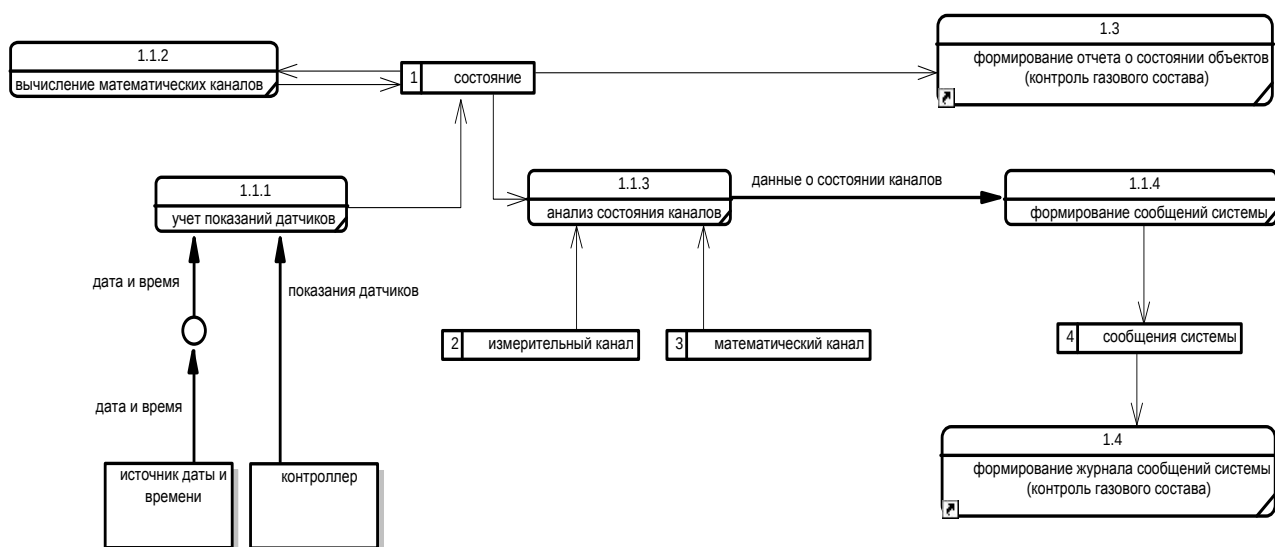
					ПГУ 1.090301.04.001 ПЗ	Лист
						91
Изм.	Лист	№ докум.	Подпись	Дата		



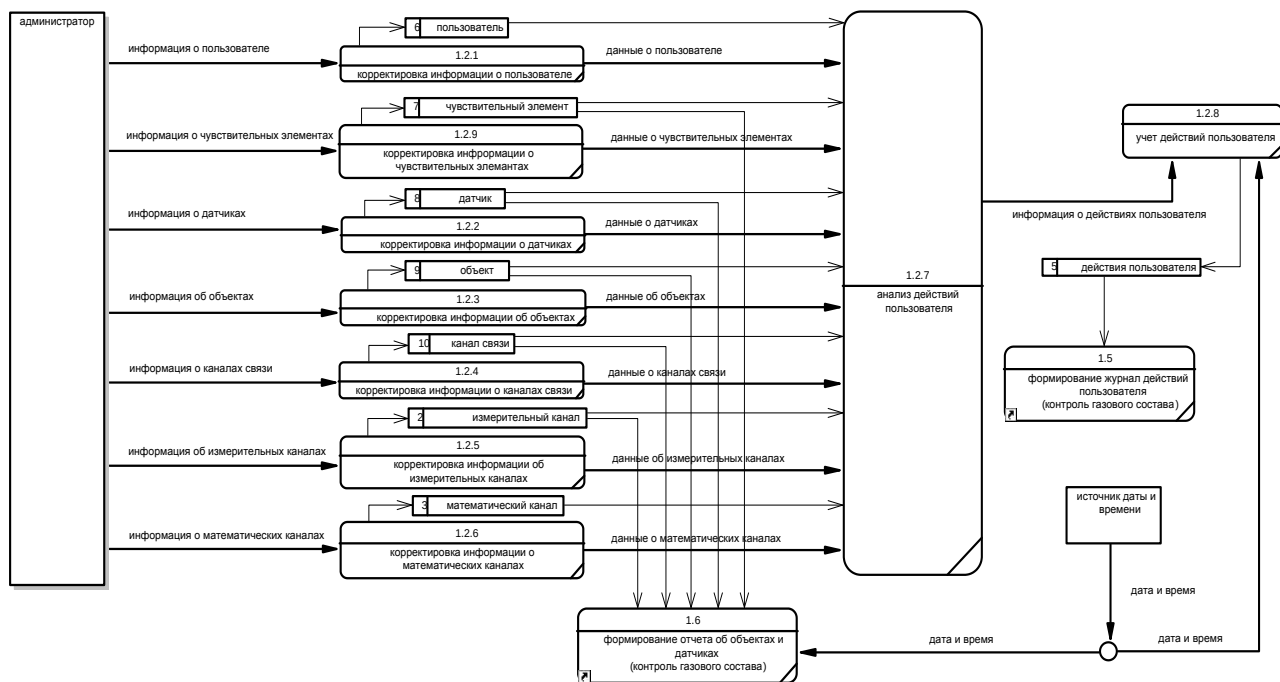
					ПГУ 1.090301.04.003 Д2			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Бычинин М.А.			Программное обеспечение АРМ оператора системы контроля газового состава Модель функциональная Диаграмма контекстная	Лит.	Лист	Листов
Провер.		Трокоз Д.А.					1	1
						ФВТ гр. 13ВВ1		
Н. Контр.		Кучин А.В.						
Утв.		Пашенко Д.В.						



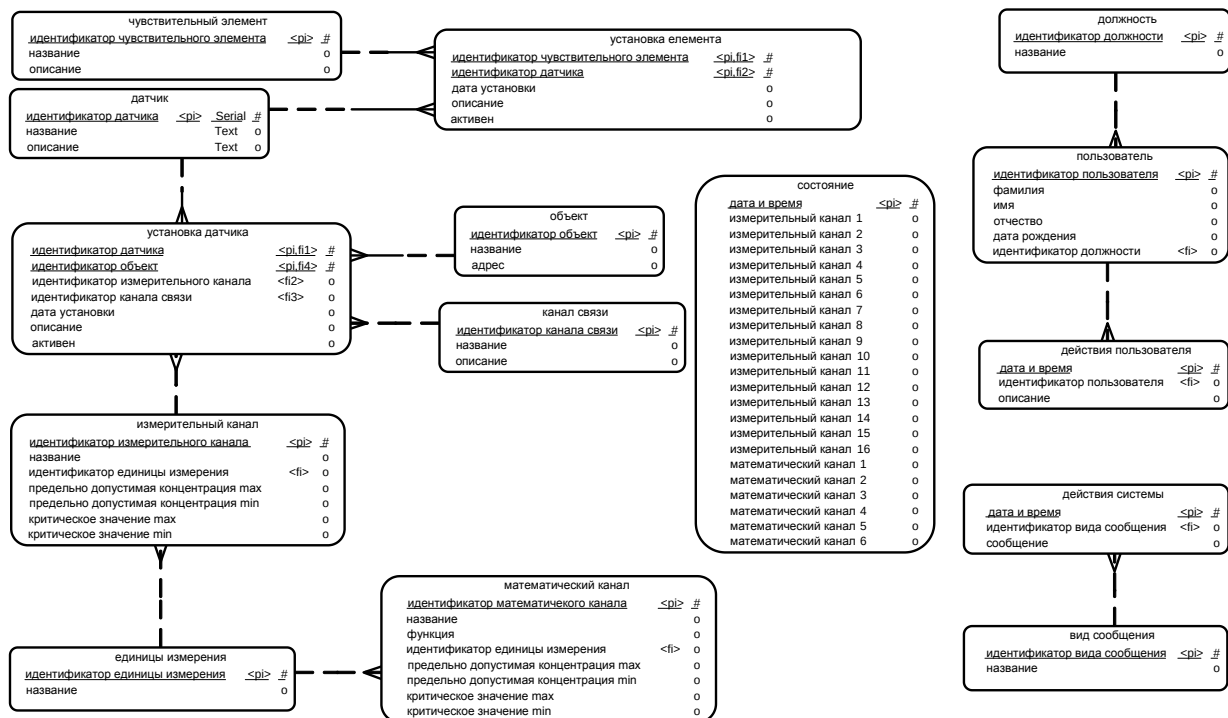
					ПГУ 1.090301.04.004 Д2			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Бычинин М.А.			Программное обеспечение АРМ оператора системы контроля газового состава Диаграмма декомпозиции Диаграмма контекстная	Лит.	Лист	Листов
Провер.		Трокоз Д.А.					1	1
						ФВТ гр. 13ВВ1		
Н. Контр.		Кучин А.В.						
Утв.		Пашенко Д.В.						



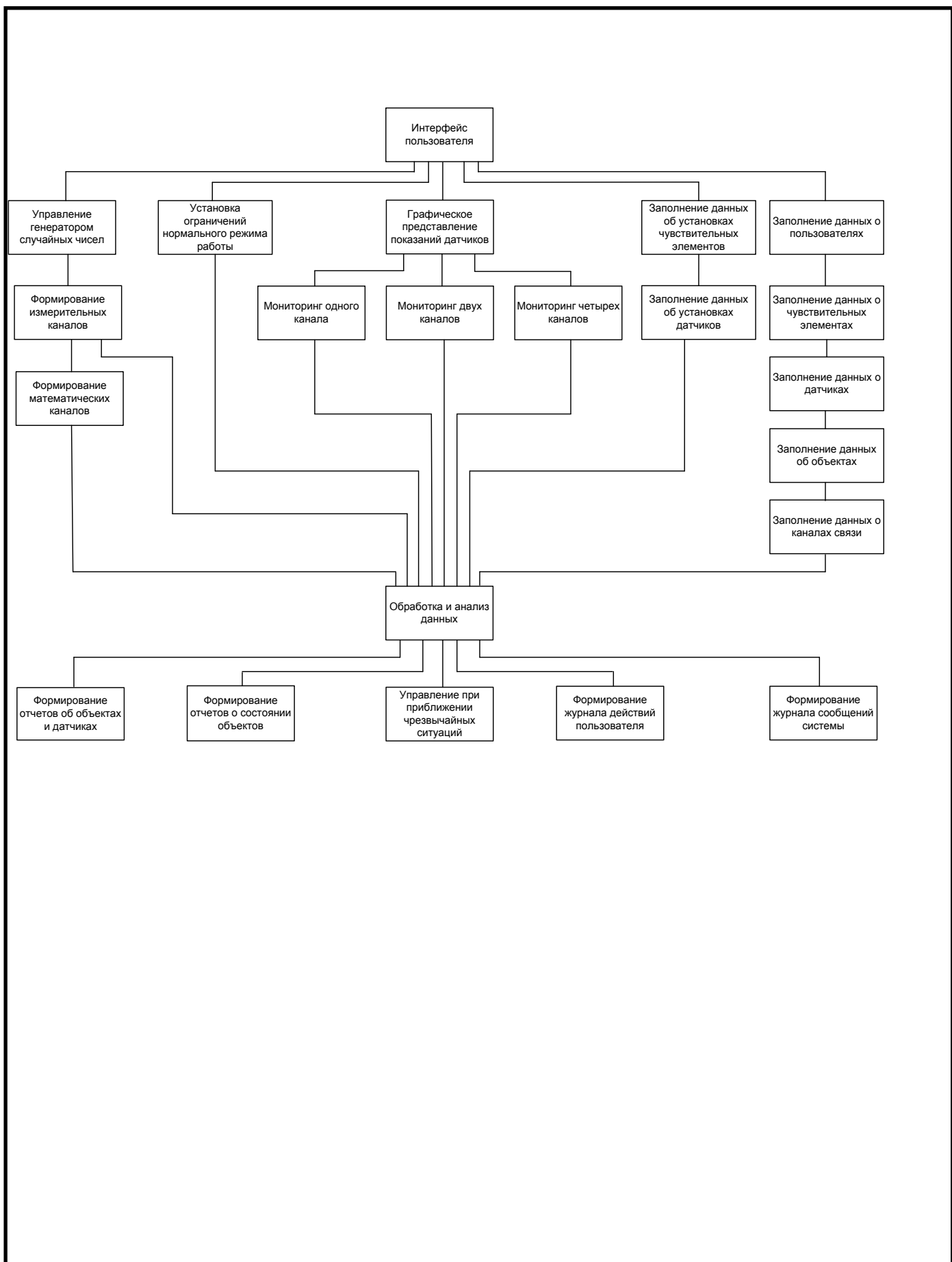
					ПГУ 1.090301.04.005 Д2		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Бычинин М.А.				Программное обеспечение АРМ оператора системы контроля газового состава Учет и анализ показаний датчиков Диаграмма декомпозиции процесса	Лит.	Лист
Провер.	Трокоз Д.А.						Листов
Н. Контр.	Кучин А.В.					1	1
Утв.	Пашенко Д.В.					ФВТ гр. 13ВВ1	



					ПГУ 1.090301.04.006 Д2		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Бычинин М.А.				Программное обеспечение АРМ оператора системы контроля газового состава Учет и анализ действий пользователя Диаграмма декомпозиции	Лит.	Лист
Провер.	Трокоз Д.А.					1	1
Н. Контр.	Кучин А.В.					ФВТ гр. 13ВВ1	
Утв.	Пашенко Д.В.						



					ПГУ 1.090301.04.007 ДЗ						
Изм.	Лист	№ докум.	Подпись	Дата	Программное обеспечение АРМ оператора системы контроля газового состава ЕР – диаграмма логического уровня						
Разраб.		Бычинин М.А									
Провер.		Трокоз Д.А.			ФВТ гр. 13ВВ1						
Н. Контр.		Кучин А.В.									
Утв.		Пашенко Д.В.									
					Лит.	Лист	Листов				
						1	1				



					ПГУ 1.090301.04.009 Д5			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Бычинин М.А.				Программное обеспечение АРМ оператора системы контроля газового состава Диаграмма взаимодействия программ	Лит.	Лист	Листов
Провер.	Трокоз Д.А.						1	1
Н. Контр.	Кучин А.В.					ФВТ гр. 13ВВ1		
Утв.	Пашенко Д.В.							

ПРИЛОЖЕНИЕ Б
(Презентация)

					ПГУ 1.090301.04.001 ПЗ	Лист
						101
Изм.	Лист	№ докум.	Подпись	Дата		

Программное обеспечение АРМ оператора системы контроля газового состава



Выполнил: студент группы 13ВВ1 Бычинин М.А.

Рисунок Б1 – Слайд 1

Цель работы – разработать для предприятия нефтегазового комплекса программное обеспечение АРМ оператора системы контроля газового состава.
Разрабатываемая система должна обеспечивать выполнение следующих функций:

- имитация сигналов с датчика;
- контроль 16 измерительных каналов;
- хранение данных о примененных датчиках и чувствительных элементах каждого из каналов;
- хранение информации об объектах, на которых установлены датчики и вида примененных каналов связи для соединения с арм;
- формирование 6 математических каналов для расчета необходимых параметров из отслеживаемых;
- установка режима нормальной работы для каждого измерительного канала;
- отслеживание состояния каждого измерительного канала в виде:
- сохранение значений измерительных каналов в базе данных;
- протоколирование действий пользователя и сообщений системы;
- вывод архивной информации.

Рисунок Б2 – Слайд 2

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.04.001 ПЗ

Лист

102



Рисунок Б3 – Слайд 3

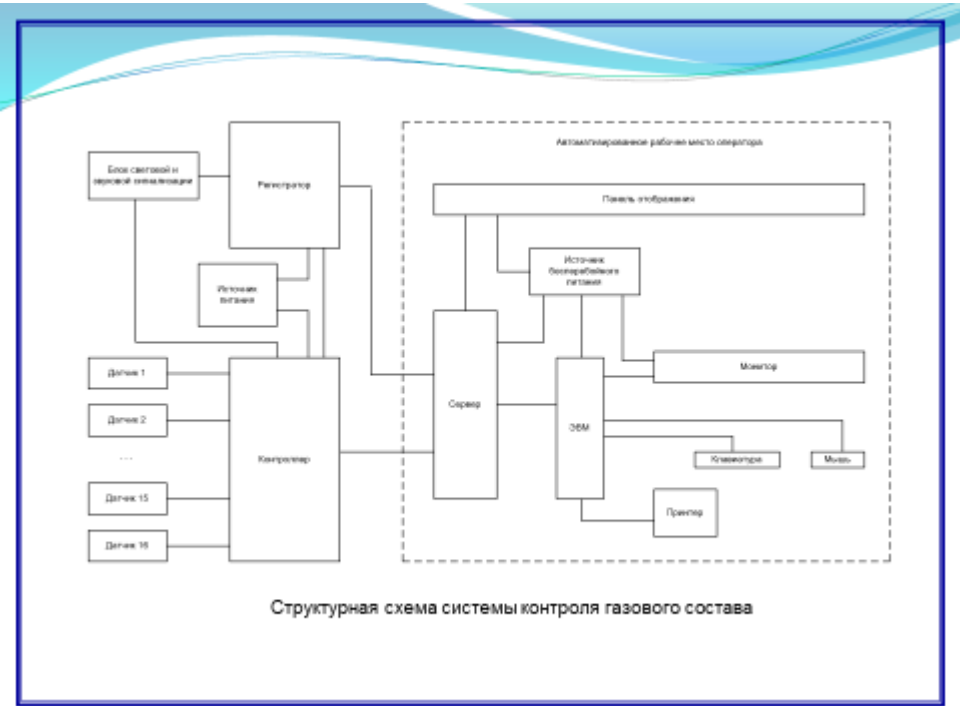


Рисунок Б4 – Слайд 4

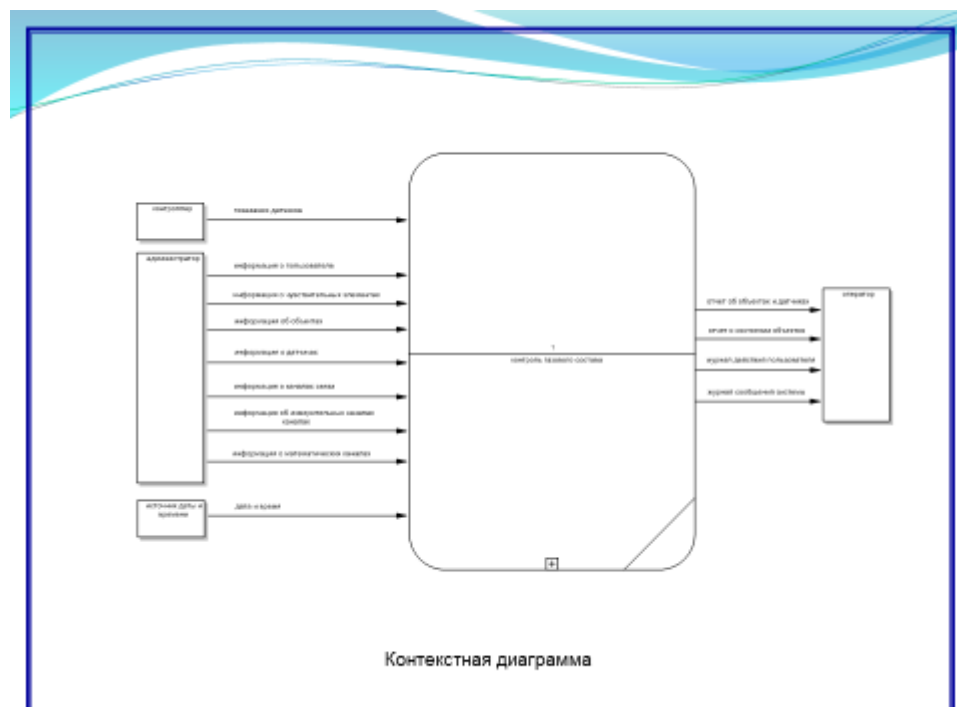
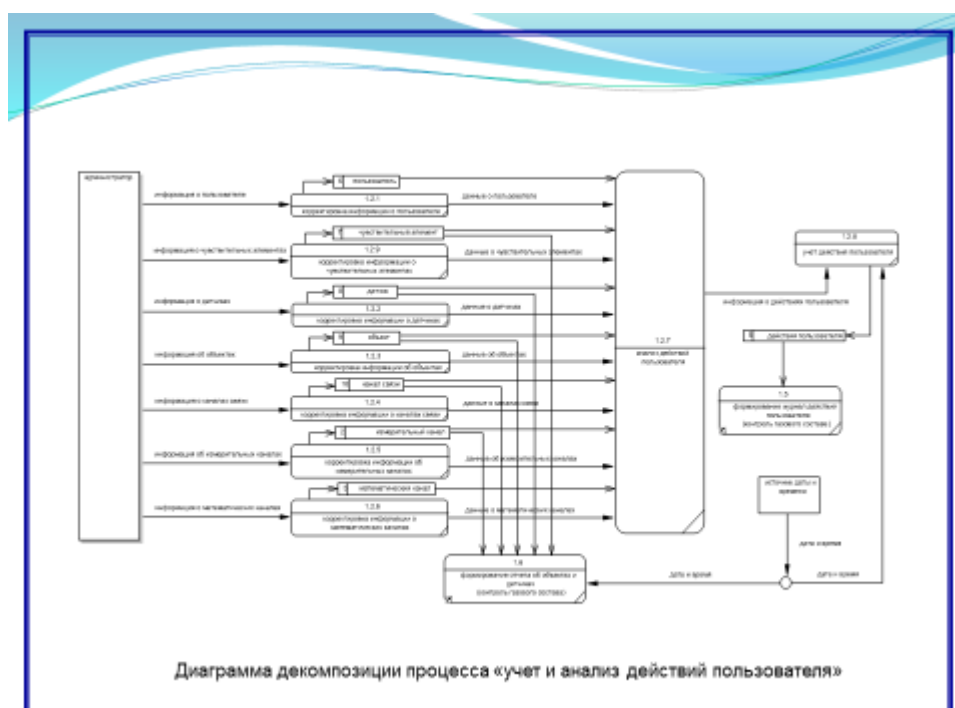


Рисунок Б5 – Слайд 5



Рисунок Б6 – Слайд 6



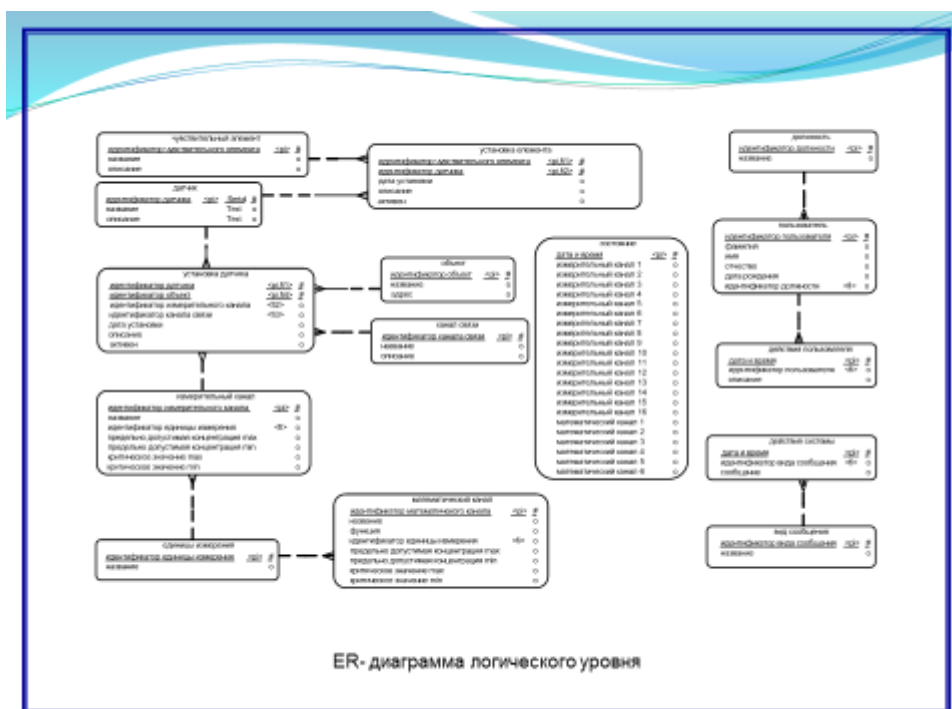


Рисунок Б9 – Слайд 9

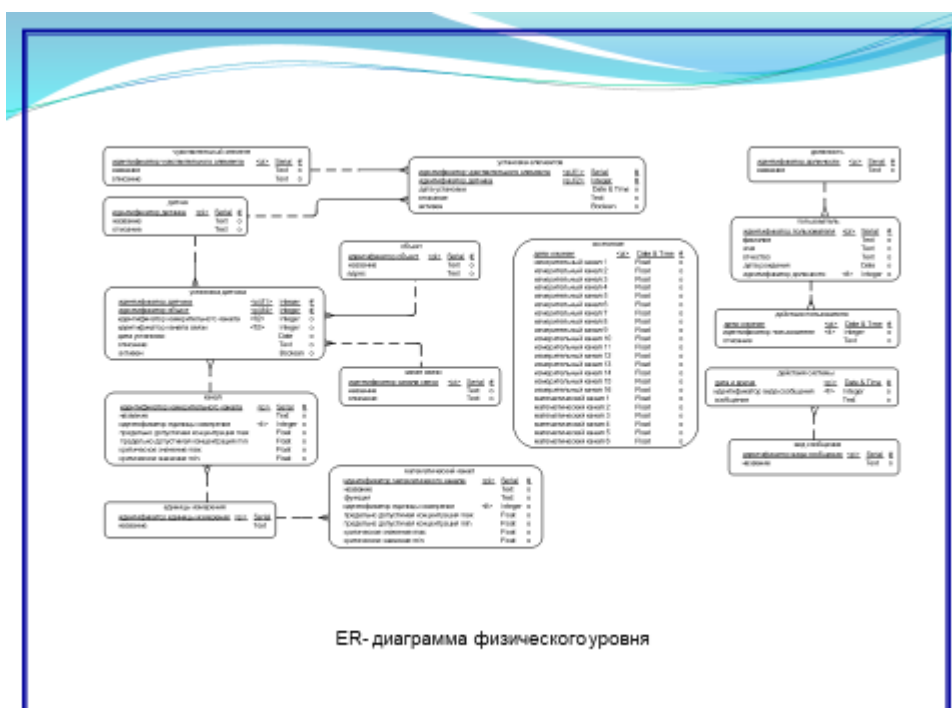


Рисунок Б10 – Слайд 10

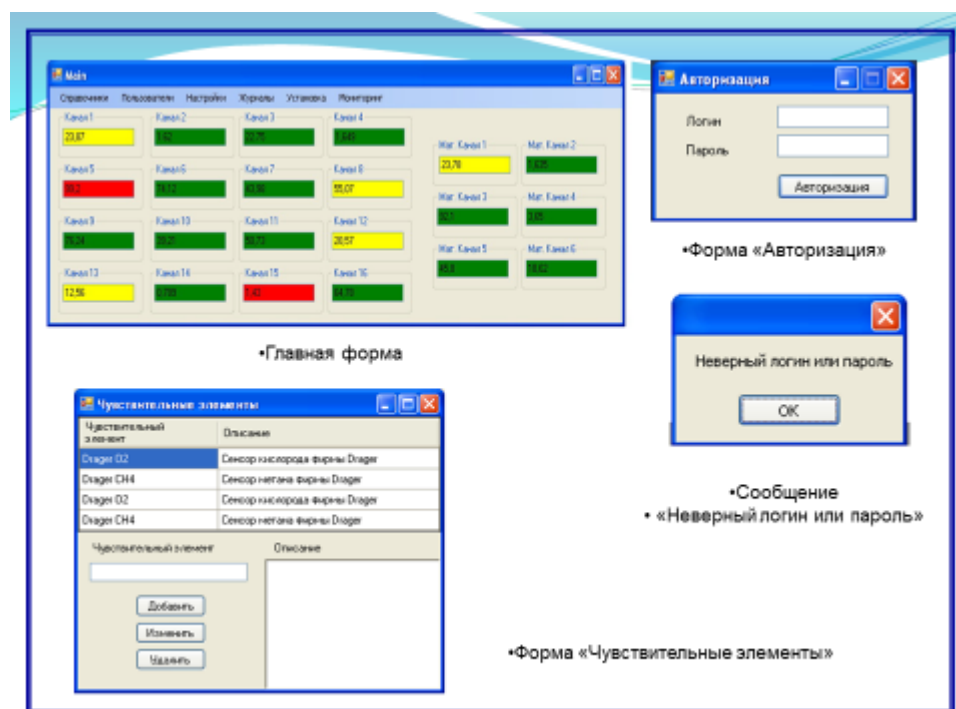


Рисунок Б11 – Слайд 11

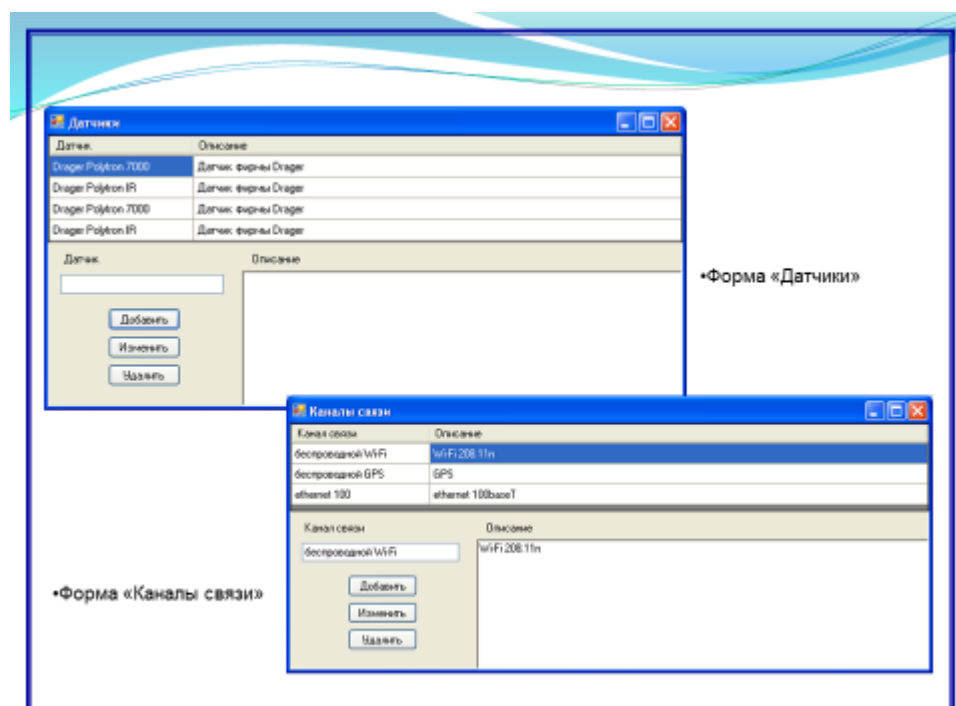


Рисунок Б12 – Слайд 12

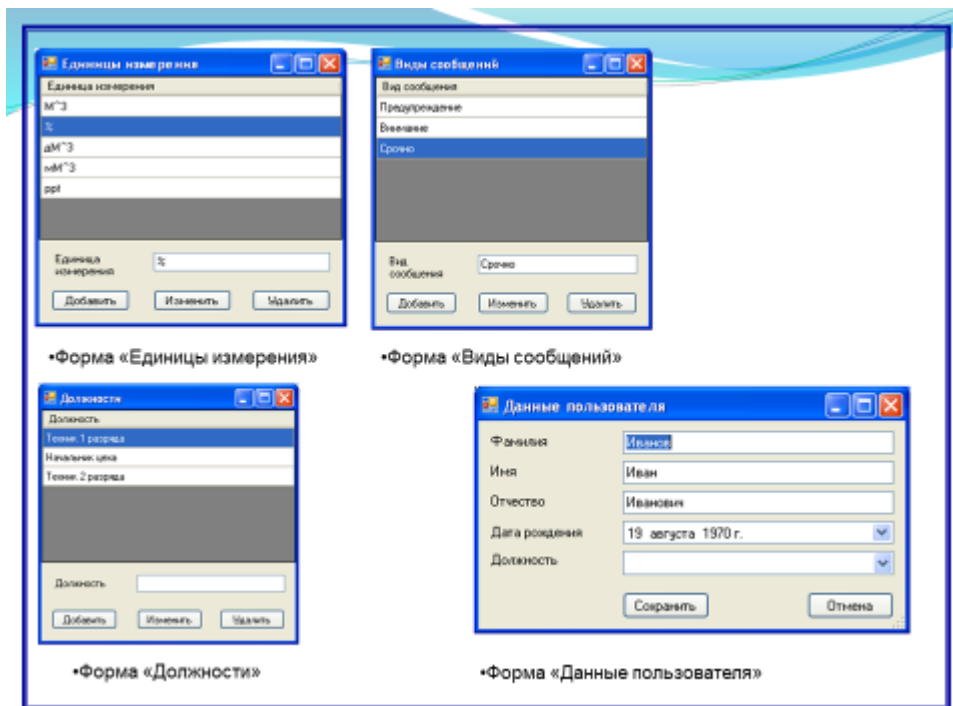


Рисунок Б13 – Слайд 13

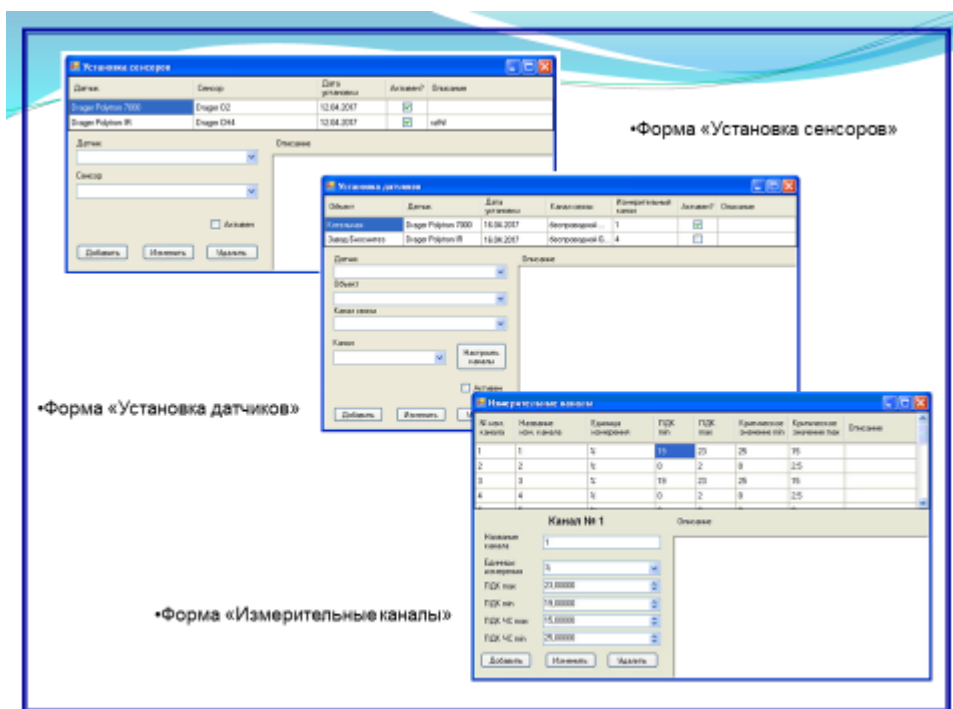
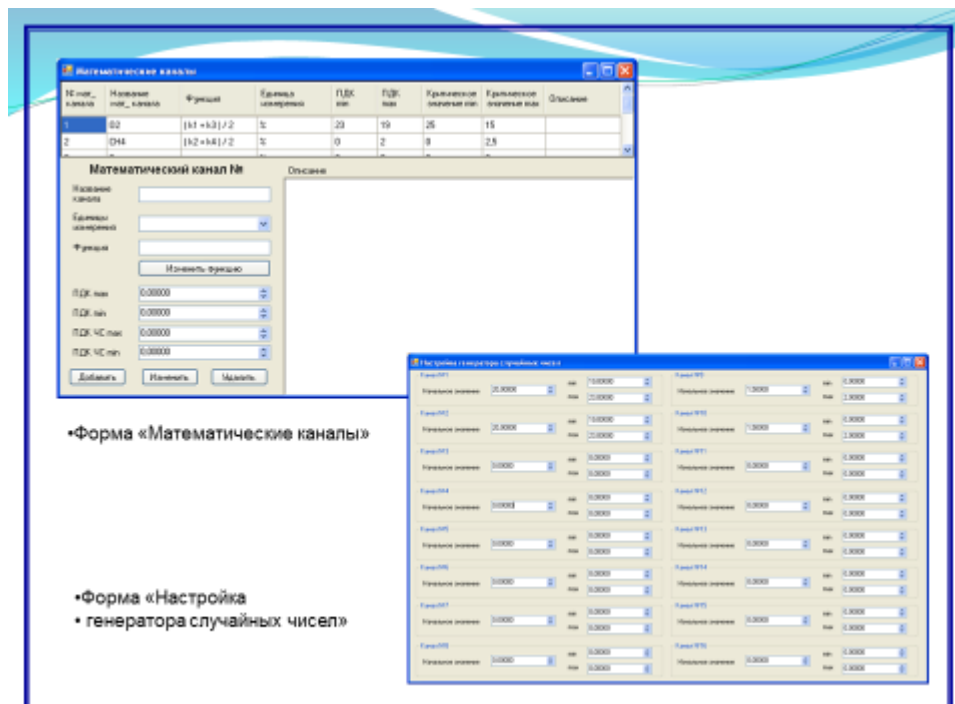
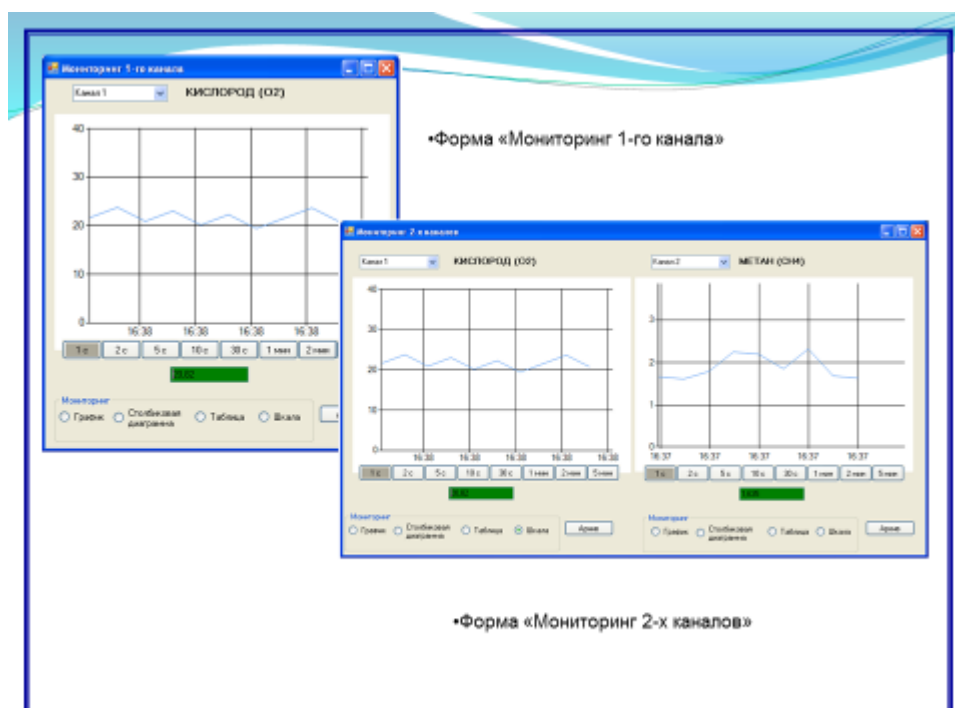


Рисунок Б14 – Слайд 14



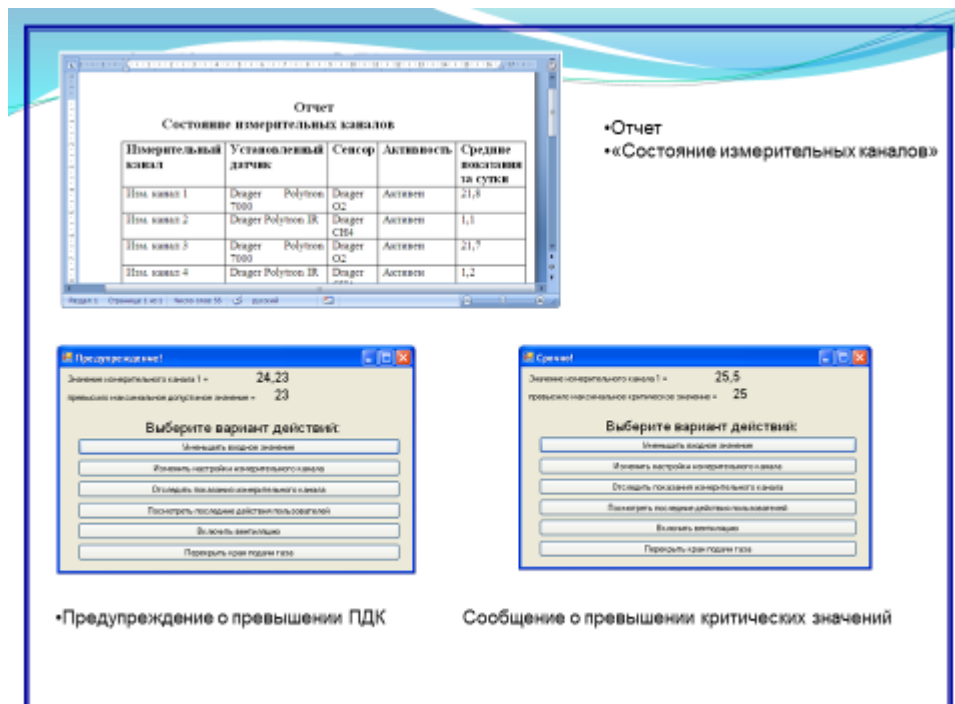


Рисунок Б17 – Слайд 17



Рисунок Б18 – Слайд 18

Приложение В
(Фрагмент текста программы)

					ПГУ 1.090301.04.001 ПЗ	Лист
						111
Изм.	Лист	№ докум.	Подпись	Дата		

Dat.cs

```
Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

Namespace monitoring
{
    Public partial class dat : Form
    {
        Public dat()
        {
            Initializecomponent();
        }
        Private dattableadapter datt = new dattableadapter();
        Private int id = 0;
        Private void dat_Load(object sender, eventargs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.dat". При необходимости она может быть перемещена или удалена.
            This.dattableadapter.Fill(this.monitoringdataset.dat);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.dat". При необходимости она может быть перемещена или удалена.
            // this.dattableadapter.Fill(this.monitoringdataset.dat);
            Datagridview1.Height = this.Height - panel1.Height + 5;
            Panel2.Width = this.Width - panel3.Width - 10;
        }
        Private void dat_Resize(object sender, eventargs e)
        {
            Datagridview1.Height = this.Height - panel1.Height + 5;
            Panel2.Width = this.Width - panel3.Width - 10;
        }
        Private void addbutton_Click(object sender, eventargs e)
        {
            If (textbox1.Text != "")
            {
                Datt.insertquery(textbox1.Text, richtextbox1.Text);
                Datt.Fill(this.monitoringdataset.dat);
                Textbox1.Clear();
                Richtextbox1.Clear();
            }
            Else
            {
                MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
                tons.OK, messageboxicon.Warning);
            }
        }
        Private void edbutton_Click(object sender, eventargs e)
        {
            Try
            {
                Int tmp = id;
                If (tmp != 0)
```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		112

```

        {
            If (messagebox.Show("Изменить запись '" +
datt.getdatabyid(tmp).Rows[0][1].toString() + "' ?", "Изменить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
            {
                If (textbox1.Text != "")
                {
                    Datt.updatequery(textbox1.Text, richtextbox1.Text, id);
                    Datt.Fill(this.monitoringdataset.dat);
                    Textbox1.Clear();
                    Richtextbox1.Clear();
                }
                Else
                {
                    MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
                }
            }
        }
        Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для из-
менения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}
Private void delbutton_Click(object sender, eventargs e)
{
    Try
    {
        Int tmp = id;
        If (tmp != 0)
        {
            If (messagebox.Show("Удалить '" +
datt.getdatabyid(tmp).Rows[0][1].toString() + "' ?", "Удалить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
            {
                Datt.deletequery(id);
                Datt.Fill(this.monitoringdataset.dat);
                Textbox1.Clear();
                Richtextbox1.Clear();
            }
        }
        Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для уда-
ления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void datagridview1_cellclick(object sender, datagridviewcelleventargs
e)
{
    Try
    {
        Id = Con-
vert.ToInt32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[0].Val
ue);

        If (id > 0)
        {
            Textbox1.Text = datt.getdatabyid(id).Rows[0][1].toString();
            Richtextbox1.Text = datt.getdatabyid(id).Rows[0][2].toString();
        }
    }
}

```

```

        Else
        {
            Textbox1.Clear();
            Richtextbox1.Clear();
        }
    }
    Catch (nullreferenceexception) { };
}
}
}

```

Dol.cs

```

Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

```

Namespace monitoring

```

{
    Public partial class dol : Form
    {
        Public dol()
        {
            Initializecomponent();
        }

        Private doltableadapter dolt = new doltableadapter();
        Private int id = 0;

        Private void dol_Load(object sender, eventargs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу "monitor-
            ingdataset.dol". При необходимости она может быть перемещена или удалена.
            This.doltableadapter.Fill(this.monitoringdataset.dol);
        }

        Private void dol_Resize(object sender, eventargs e)
        {
            Datagridview1.Height = this.Height - panel1.Height + 5;
        }

        Private void addbutton_Click(object sender, eventargs e)
        {
            If (textbox1.Text != "")
            {
                Dolt.insertquery(textbox1.Text);
                Dolt.Fill(this.monitoringdataset.dol);
                Textbox1.Clear();
            }
            Else
            {
                MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
                tons.OK, messageboxicon.Warning);
            }
        }
    }
}

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		114

```

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = id;
        If (tmp != 0)
        {
            If (MessageBox.Show("Изменить '" +
dolt.getdatabyid(tmp).Rows[0][1].ToString() + "' ?", "Изменить запись? ", MessageBoxButtons.YesNo, MessageBoxIcon.Stop) == DialogResult.Yes)
            {
                If (textBox1.Text != "")
                {
                    Dolt.updatequery(textBox1.Text, id);
                    Dolt.Fill(this.monitoringdataset.dol);
                    TextBox1.Clear();
                }
                Else
                {
                    MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, MessageBoxIcon.Warning);
                }
            }
            Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, MessageBoxIcon.Error);
        }
        Catch (NullReferenceException) { MessageBox.Show("Выберите запись для
изменения", "Ошибка!", messageboxbuttons.OK, MessageBoxIcon.Error); }
    }

Private void delbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = id;
        If (tmp != 0)
        {
            If (MessageBox.Show("Удалить запись '" +
dolt.getdatabyid(tmp).Rows[0][1].ToString() + "' ?", "Удалить запись? ", MessageBoxButtons.YesNo, MessageBoxIcon.Stop) == DialogResult.Yes)
            {
                Dolt.deletequery(id);
                Dolt.Fill(this.monitoringdataset.dol);
                TextBox1.Clear();
            }
            Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, MessageBoxIcon.Error);
        }
        Catch (NullReferenceException) { MessageBox.Show("Выберите запись для
удаления", "Ошибка!", messageboxbuttons.OK, MessageBoxIcon.Error); }
    }

Private void datagridview1_cellclick(object sender, DataGridViewCellEventArgs
e)
{
    Try
    {

```

```

        Id = Convert.ToInt32(this.datagridview1.Rows[this.datagridview1.CurrentRow.Index].Cells[0].Value);
        If (id > 0)
        {
            TextBox1.Text = dolt.getdatabyid(id).Rows[0][1].ToString();
        }
        Else
        {
            TextBox1.Clear();
        }
    }
    Catch (NullReferenceException) { };
}
}
}Ob.cs

```

```

Using System;
Using System.Collections.Generic;
Using System.ComponentModel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

```

```

Namespace monitoring
{
    Public partial class ob : Form
    {
        Public ob()
        {
            InitializeComponent();

            Private objectstableadapter obt = new objectstableadapter();
            Private int id = 0;

            Private void ob_Resize(object sender, EventArgs e)
            {
                DataGridView1.Height = this.Height - panel1.Height + 5;
                Panel2.Width = this.Width - panel3.Width - 10;
            }

            Private void ob_Load(object sender, EventArgs e)
            {
                // TODO: данная строка кода позволяет загрузить данные в таблицу "monitoringdataset.objects". При необходимости она может быть перемещена или удалена.
                This.objectstableadapter.Fill(this.monitoringdataset.objects);

                DataGridView1.Height = this.Height - panel1.Height + 5;
                Panel2.Width = this.Width - panel3.Width - 10;
            }

            Private void datagridview1_CellClick(object sender, DataGridViewCellEventArgs e)
            {
                Try
                {

```

```

        Id = Convert.ToInt32(this.dataGridView1.Rows[this.dataGridView1.CurrentRow.Index].Cells[0].Value);

        If (id > 0)
        {
            TextBox1.Text = obt.GetDataById(id).Rows[0][1].ToString();
            TextBox2.Text = obt.GetDataById(id).Rows[0][2].ToString();
            TextBox3.Text = obt.GetDataById(id).Rows[0][3].ToString();
            RichTextBox1.Text = obt.GetDataById(id).Rows[0][4].ToString();
        }
        Else
        {
            TextBox1.Clear();
            TextBox2.Clear();
            TextBox3.Clear();
            RichTextBox1.Clear();
        }
    }
    Catch (NullReferenceException) { };
}

Private void addButton_Click(object sender, EventArgs e)
{
    If (textBox1.Text != "")
    {
        Obt.InsertQuery(textBox1.Text, textBox2.Text, textBox3.Text, richTextBox1.Text);
        Obt.Fill(this.monitoringdataset.objects);
        TextBox1.Clear();
        TextBox2.Clear();
        TextBox3.Clear();
        RichTextBox1.Clear();
    }
    Else
    {
        MessageBox.Show("Есть незаполненные поля", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = id;
        If (tmp != 0)
        {
            If (MessageBox.Show("Изменить запись '" + obt.GetDataById(tmp).Rows[0][1].ToString() + "' ?", "Изменить запись?", MessageBoxButtons.YesNo, MessageBoxIcon.Stop) == DialogResult.Yes)
            {
                If (textBox1.Text != "")
                {
                    Obt.UpdateQuery(textBox1.Text, textBox2.Text, textBox3.Text, richTextBox1.Text, id);
                    Obt.Fill(this.monitoringdataset.objects);
                    TextBox1.Clear();
                    TextBox2.Clear();
                    TextBox3.Clear();
                    RichTextBox1.Clear();
                }
            }
            Else

```

```

        {
            MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
        }
    }
    Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
}
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для
изменения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void delbutton_Click(object sender, eventargs e)
{
    Try
    {
        Int tmp = id;
        If (tmp != 0)
        {
            If (messagebox.Show("Удалить запись '" +
obt.getdatabyid(tmp).Rows[0][1].toString() + "' ?", "Удалить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
            {
                Obt.deletequery(id);
                Obt.Fill(this.monitoringdataset.objects);
                Textbox1.Clear();
                Textbox2.Clear();
                Textbox3.Clear();
                Richtextbox1.Clear();
            }
        }
        Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для
удаления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}
}
}

```

K.cs

```

Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

```

```

Namespace monitoring
{
    Public partial class k : Form
    {
        Public k()
        {
            Initializecomponent();
        }
    }
}

```

```

Private e_itableadapter eit = new e_itableadapter();
Private ktableadapter kt = new ktableadapter();
Private int idk = 0;
Private int idei = 0;

Private void k_Resize(object sender, eventargs e)
{
    Datagridview1.Height = this.Height - panel1.Height - 35;
    Panel2.Width = this.Width - panel3.Width - 10;
}

Private void k_Load(object sender, eventargs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу
"monitoringdataset.kan". При необходимости она может быть перемещена или удалена.
    This.kantableadapter.Fill(this.monitoringdataset.kan);
    // TODO: данная строка кода позволяет загрузить данные в таблицу
"monitoringdataset.e_i". При необходимости она может быть перемещена или удалена.
    This.e_itableadapter.Fill(this.monitoringdataset.e_i);

    // datagridview1.Height = this.Height - panel1.Height + 5;
    Datagridview1.Height = this.Height - panel1.Height - 35;
    Panel2.Width = this.Width - panel3.Width - 10;

    Label8.Text = "Канал № ";
    Combobox1.resettext();
    Numericupdown1.Value = 0;
    Textbox1.Clear();
    Numericupdown2.Value = 0;
    Numericupdown3.Value = 0;
    Numericupdown4.Value = 0;
    Richtextbox1.Clear();
}

Private void datagridview1_cellclick(object sender, datagridviewcelleventargs
e)
{
    Try
    {
        Idk = Convert.ToInt32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[0].Value);

        Try
        {
            Idei = Convert.ToInt32(kt.getdatabyid(idk).Rows[0][2]);
        }
        Catch (invalidcastexception){};

        If (idk > 0)
        {
            Label8.Text = "Канал № " + idk;
            Textbox1.Text = kt.getdatabyid(idk).Rows[0][1].ToString();
            Try
            {
                Combobox1.selectedvalue = Convert.ToInt32(kt.getdatabyid(idk).Rows[0][2]);
            }
            Catch (invalidcastexception) { };
            Numericupdown1.Value = (decimal)Convert.ToDouble(kt.getdatabyid(idk).Rows[0][3]);
            Numericupdown2.Value = (decimal)Convert.ToDouble(kt.getdatabyid(idk).Rows[0][4]);
        }
    }
}

```

```

        Numericupdown3.Value = (decimal)Convert.ToDouble(kit.getdatabyid(idk).Rows[0][5]);
        Numericupdown4.Value = (decimal)Convert.ToDouble(kit.getdatabyid(idk).Rows[0][6]);
        Richtextbox1.Text = kit.getdatabyid(idk).Rows[0][7].ToString();
    }
    Else
    {
        Label8.Text = "Канал № ";
        Combobox1.ResetText();
        Numericupdown1.Value = 0;
        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }
}
Catch (NullReferenceException) { };
}

Private void addButton_Click(object sender, EventArgs e)
{
    If (textbox1.Text != "")
    {
        Int ei = 0;
        Try
        {
            Ei = Convert.ToInt32(combobox1.SelectedValue);
            If (ei != 0)
            {
                Kit.InsertQuery(textbox1.Text, ei, Convert.ToDouble(numericupdown1.Value), Convert.ToDouble(numericupdown2.Value), Convert.ToDouble(numericupdown3.Value), Convert.ToDouble(numericupdown4.Value), richtextbox1.Text);
            }
            Else
            {
                Kit.InsertQuery(textbox1.Text, null, Convert.ToDouble(numericupdown1.Value), Convert.ToDouble(numericupdown2.Value), Convert.ToDouble(numericupdown3.Value), Convert.ToDouble(numericupdown4.Value), richtextbox1.Text);
            }
        }
        Catch (NullReferenceException) { kit.InsertQuery(textbox1.Text, null, Convert.ToDouble(numericupdown1.Value), Convert.ToDouble(numericupdown2.Value), Convert.ToDouble(numericupdown3.Value), Convert.ToDouble(numericupdown4.Value), richtextbox1.Text); };

        Kit.Fill(this.monitoringdataset.k);
        Label8.Text = "Канал № ";
        Combobox1.ResetText();
        Numericupdown1.Value = 0;
        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }
    Else
    {

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		120

```

        MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
tons.OK, messageboxicon.Warning);
    }
}

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = idk;

        If (tmp != 0)
        {
            If (messagebox.Show("Изменить запись '" +
kt.getdatabyid(tmp).Rows[0][1].ToString() + "' ?", "Изменить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == DialogResult.Yes)
            {
                If (textbox1.Text != "")
                {
                    Try
                    {
                        Int ei = Convert.ToInt32(ComboBox1.SelectedValue);
                        If (ei != 0)
                        {
                            Kt.updatequery(textbox1.Text, ei, Con-
vert.ToDouble(NumericUpDown1.Value), Convert.ToDouble(NumericUpDown2.Value), Con-
vert.ToDouble(NumericUpDown3.Value), Convert.ToDouble(NumericUpDown4.Value), richtext-
box1.Text, idk);
                        }
                        Else
                        {
                            Kt.updatequery(textbox1.Text, null, Con-
vert.ToDouble(NumericUpDown1.Value), Convert.ToDouble(NumericUpDown2.Value), Con-
vert.ToDouble(NumericUpDown3.Value), Convert.ToDouble(NumericUpDown4.Value), richtext-
box1.Text, idk);
                        }
                    }
                    Catch (NullReferenceException) {
kt.updatequery(textbox1.Text, null, Convert.ToDouble(NumericUpDown1.Value), Con-
vert.ToDouble(NumericUpDown2.Value), Convert.ToDouble(NumericUpDown3.Value), Con-
vert.ToDouble(NumericUpDown4.Value), richTextBox1.Text, idk); };
                    Kt.Fill(this.monitoringdataset.k);
                    Label8.Text = "Канал № ";
                    ComboBox1.ResetText();
                    NumericUpDown1.Value = 0;
                    TextBox1.Clear();
                    NumericUpDown2.Value = 0;
                    NumericUpDown3.Value = 0;
                    NumericUpDown4.Value = 0;
                    RichTextBox1.Clear();
                }
            }
            Else
            {
                MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
            }
        }
    }
    Else
    {
        Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
}

```

```

        Catch (nullreferenceexception) { messagebox.Show("Выберите запись для из-
менения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
    }

    Private void delbutton_Click(object sender, eventargs e)
    {
        Try
        {
            Int tmp = idk;

            If (tmp != 0)
            {
                If (messagebox.Show("Удалить запись?", "Удалить запись? ", mes-
sageboxbuttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
                {
                    Kt.deletequery(idk);
                    Kt.Fill(this.monitoringdataset.k);
                    Label8.Text = "Канал № ";
                    Combobox1.resettext();
                    Numericupdown1.Value = 0;
                    Textbox1.Clear();
                    Numericupdown2.Value = 0;
                    Numericupdown3.Value = 0;
                    Numericupdown4.Value = 0;
                    Richtextbox1.Clear();
                }
            }
            Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
        }
        Catch (nullreferenceexception) { messagebox.Show("Выберите запись для уда-
ления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
    }
}

```

Mk.cs

```

Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

```

```

Namespace monitoring
{
    Public partial class mk : Form
    {
        Public mk()
        {
            Initializecomponent();
        }

        Static private znach1 func;
        Static public znach1 Func
        {

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		122

```

        Get { return func; }
        Set { func = value; }
    }

    Private e_itableadapter eit = new e_itableadapter();
    Private m_ktableadapter mkt = new m_ktableadapter();
    Private int idmk = 0;
    Private int idei = 0;

    Private void mk_Resize(object sender, eventargs e)
    {
        // datagridview1.Height = this.Height - panel1.Height + 5;
        Ddatagridview1.Height = this.Height - panel1.Height - 35;
        Panel2.Width = this.Width - panel3.Width - 10;
    }

    Private void mk_Load(object sender, eventargs e)
    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "monitoringdataset.mkan". При необходимости она может быть перемещена или удалена.
        This.mkantableadapter.Fill(this.monitoringdataset.mkan);
        // TODO: данная строка кода позволяет загрузить данные в таблицу
        "monitoringdataset.e_i". При необходимости она может быть перемещена или удалена.
        This.e_itableadapter.Fill(this.monitoringdataset.e_i);

        // datagridview1.Height = this.Height - panel1.Height + 5;
        Ddatagridview1.Height = this.Height - panel1.Height - 35;
        Panel2.Width = this.Width - panel3.Width - 10;

        Label8.Text = "Математический канал № ";
        Combobox1.resettext();
        Numericupdown1.Value = 0;
        Textbox1.Clear();
        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }

    Private void datagridview1_cellclick(object sender, datagridviewcelleventargs
e)
    {
        Try
        {
            Idmk = Con-
vert.toint32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[0].Val
ue);

            Try
            {
                Idei = Convert.toint32(mkt.getdatabyid(idmk).Rows[0][3]);
            }
            Catch (invalidcastexception) { };
            If (idmk > 0)
            {
                Label8.Text = "Математический канал № " + idmk;

                Textbox1.Text = mkt.getdatabyid(idmk).Rows[0][1].toString();
                Textbox2.Text = mkt.getdatabyid(idmk).Rows[0][2].toString();
                Try

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		123

```

        {
            Combobox1.selectedvalue = Convert.ToInt32(mkt.getdatabyid(idmk).Rows[0][3]);
        }
        Catch (invalidcastexception) { };
        Numericupdown1.Value = (decimal)Convert.ToDouble(mkt.getdatabyid(idmk).Rows[0][4]);
        Numericupdown2.Value = (decimal)Convert.ToDouble(mkt.getdatabyid(idmk).Rows[0][5]);
        Numericupdown3.Value = (decimal)Convert.ToDouble(mkt.getdatabyid(idmk).Rows[0][6]);
        Numericupdown4.Value = (decimal)Convert.ToDouble(mkt.getdatabyid(idmk).Rows[0][7]);
        Richtextbox1.Text = mkt.getdatabyid(idmk).Rows[0][8].ToString();
    }
    Else
    {
        Label8.Text = "Математический канал № ";
        Combobox1.resettext();
        Numericupdown1.Value = 0;
        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }
}
Catch (nullreferenceexception) { };
}

Private void fun_Click(object sender, EventArgs e)
{
    F f = new f();
    Znach MK = new znach();
    MK.T = idmk;
    F.mkn = MK;
    F.Show();
}

Private void addbutton_Click(object sender, EventArgs e)
{
    If (textbox1.Text != "")
    {
        Try
        {
            Int ei = Convert.ToInt32(combobox1.selectedvalue);
            If (ei != 0)
            {
                Mkt.insertquery(textbox1.Text, textbox2.Text, ei, Convert.ToDouble(numericupdown1.Value), Convert.ToDouble(numericupdown2.Value), Convert.ToDouble(numericupdown3.Value), Convert.ToDouble(numericupdown4.Value), richtextbox1.Text);
            }
            Else
            {
                Mkt.insertquery(textbox1.Text, textbox2.Text, null, Convert.ToDouble(numericupdown1.Value), Convert.ToDouble(numericupdown2.Value), Convert.ToDouble(numericupdown3.Value), Convert.ToDouble(numericupdown4.Value), richtextbox1.Text);
            }
        }
    }
}
}

```

```

        Catch (nullreferenceexception) { mkt.insertquery(textbox1.Text, text-
box2.Text, null, Convert.todouble(numericupdown1.Value), Con-
vert.todouble(numericupdown2.Value), Convert.todouble(numericupdown3.Value), Con-
vert.todouble(numericupdown4.Value), richtextbox1.Text); };
        Mkt.Fill(this.monitoringdataset.m_k);
        Label8.Text = "Математический канал № ";
        Combobox1.resettext();
        Numericupdown1.Value = 0;
        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }
    Else
    {
        MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
tons.OK, messageboxicon.Warning);
    }
}

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = idmk;

        If (tmp != 0)
        {
            If (messagebox.Show("Изменить запись '" +
mkt.getdatabyid(tmp).Rows[0][1].ToString() + "' ?", "Изменить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == DialogResult.Yes)
            {
                If (textbox1.Text != "")
                {
                    Try
                    {
                        Int ei = Convert.ToInt32(combobox1.selectedvalue);
                        If (ei != 0)
                        {
                            Mkt.updatequery(textbox1.Text, textbox2.Text, ei,
Convert.todouble(numericupdown1.Value), Convert.todouble(numericupdown2.Value), Con-
vert.todouble(numericupdown3.Value), Convert.todouble(numericupdown4.Value), richtext-
box1.Text, idmk);
                        }
                        Else
                        {
                            Mkt.updatequery(textbox1.Text, textbox2.Text,
null, Convert.todouble(numericupdown1.Value), Convert.todouble(numericupdown2.Value),
Convert.todouble(numericupdown3.Value), Convert.todouble(numericupdown4.Value), richtext-
box1.Text, idmk);
                        }
                    }
                    Catch (nullreferenceexception) {
mkt.updatequery(textbox1.Text, textbox2.Text, null, Con-
vert.todouble(numericupdown1.Value), Convert.todouble(numericupdown2.Value), Con-
vert.todouble(numericupdown3.Value), Convert.todouble(numericupdown4.Value), richtext-
box1.Text, idmk); };

                    Mkt.Fill(this.monitoringdataset.m_k);
                    Label8.Text = "Математический канал № ";
                    Combobox1.resettext();
                    Numericupdown1.Value = 0;

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		125

```

        Textbox1.Clear();
        Numericupdown2.Value = 0;
        Numericupdown3.Value = 0;
        Numericupdown4.Value = 0;
        Richtextbox1.Clear();
    }
    Else
    {
        MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
    }
}
Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
}
Catch (nullreferenceexception) { messagebox.Show("Выберите запись для из-
менения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void delbutton_Click(object sender, eventargs e)
{
    Try
    {
        Int tmp = idmk;
        If (tmp != 0)
        {
            If (messagebox.Show("Удалить запись?", "Удалить запись? ", mes-
sageboxbuttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
            {
                Mkt.deletequery(idmk);
                Mkt.Fill(this.monitoringdataset.m_k);
                Label8.Text = "Математический канал № ";
                Combobox1.resettext();
                Numericupdown1.Value = 0;
                Textbox1.Clear();
                Numericupdown2.Value = 0;
                Numericupdown3.Value = 0;
                Numericupdown4.Value = 0;
                Richtextbox1.Clear();
            }
        }
        Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для уда-
ления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}
}
}

```

Yst_dat.cs

```

Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;

```

					ПГУ 1.090301.04.001 ПЗ	Лист
						126
Изм.	Лист	№ докум.	Подпись	Дата		

```

Using monitoring.monitoringdatasettableadapters;

Namespace monitoring
{
    Public partial class yst_dat : Form
    {
        Public yst_dat()
        {
            Initializecomponent();
        }

        Private yst_dattableadapter yst_datt = new yst_dattableadapter();
        Private int iddat = 0;
        Private int idob = 0;
        Private bool status = false;

        Private void yst_dat_Load(object sender, eventargs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.y_dat". При необходимости она может быть перемещена или удалена.
            This.y_dattableadapter.Fill(this.monitoringdataset.y_dat);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.k". При необходимости она может быть перемещена или удалена.
            This.ktableadapter.Fill(this.monitoringdataset.k);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.ks". При необходимости она может быть перемещена или удалена.
            This.kstableadapter.Fill(this.monitoringdataset.ks);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.objects". При необходимости она может быть перемещена или удалена.
            This.objectstableadapter.Fill(this.monitoringdataset.objects);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.dat". При необходимости она может быть перемещена или удалена.
            This.dattableadapter.Fill(this.monitoringdataset.dat);

            //datagridview1.Height = this.Height - panel1.Height + 5;
            Ddatagridview1.Height = this.Height - panel1.Height - 35;
            Panel2.Width = this.Width - panel3.Width - 10;

            Combobox1.resettext();
            Combobox2.resettext();
            Combobox3.resettext();
            Combobox4.resettext();
            Checkbox1.Checked = false;
            Datetimepicker1.Value = datetime.Now;
            Richtextbox1.Clear();
        }

        Private void yst_dat_Resize(object sender, eventargs e)
        {
            //datagridview1.Height = this.Height - panel1.Height + 5;
            Ddatagridview1.Height = this.Height - panel1.Height - 35;
            Panel2.Width = this.Width - panel3.Width - 10;
        }

        Private void datagridview1_cellclick(object sender, datagridviewcelleventargs
e)
        {
            Try
            {
                Iddat = Con-
vert.toint32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[0].Val
ue);
            }
            Catch { }
        }
    }
}

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		127

```

        Idob = Convert.ToInt32(this.dataGridView1.Rows[this.dataGridView1.CurrentRow.Index].Cells[1].Value);
        If ((iddat > 0) & ((idob > 0)))
        {
            Combobox1.SelectedValue = Convert.ToInt32(yst_datt.GetDataById(idob, iddat).Rows[0][0]);
            Combobox2.SelectedValue = Convert.ToInt32(yst_datt.GetDataById(idob, iddat).Rows[0][1]);
            Combobox3.SelectedValue = Convert.ToInt32(yst_datt.GetDataById(idob, iddat).Rows[0][3]);
            Combobox4.SelectedValue = Convert.ToInt32(yst_datt.GetDataById(idob, iddat).Rows[0][4]);
            DateTimePicker1.Value = Convert.ToDateTime(yst_datt.GetDataById(idob, iddat).Rows[0][2]);
            Checkbox1.Checked = Convert.ToBoolean(yst_datt.GetDataById(idob, iddat).Rows[0][5]);
            RichTextBox1.Text = yst_datt.GetDataById(idob, iddat).Rows[0][6].ToString();
        }
        Else
        {
            Combobox1.ResetText();
            Combobox2.ResetText();
            Combobox3.ResetText();
            Combobox4.ResetText();
            Checkbox1.Checked = false;
            DateTimePicker1.Value = DateTime.Now;
            RichTextBox1.Clear();
        }
    }
    Catch (NullReferenceException) { };
}

Private void checkbox1_CheckedChanged(object sender, EventArgs e)
{
    Status = checkbox1.Checked;
}

Private void addButton_Click(object sender, EventArgs e)
{
    If ((comboBox1.SelectedValue != null) & (comboBox2.SelectedValue != null))
    {
        Int kan=0;

        Try { kan = Convert.ToInt32(comboBox4.SelectedValue); }
        Catch (NullReferenceException) { kan = 17; };
        Yst_datt.InsertQuery(Convert.ToInt32(comboBox1.SelectedValue), Convert.ToInt32(comboBox2.SelectedValue), DateTimePicker1.Value, Convert.ToInt32(comboBox3.SelectedValue), kan, checkbox1.Checked, RichTextBox1.Text);
        Yst_datt.Fill(this.monitoringDataSet.yst_dat);
        Combobox1.ResetText();
        Combobox2.ResetText();
        Combobox3.ResetText();
        Combobox4.ResetText();
        Checkbox1.Checked = false;
        DateTimePicker1.Value = DateTime.Now;
        RichTextBox1.Clear();
    }
    Else
    {

```

```

        MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
tons.OK, messageboxicon.Warning);
    }
}

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = iddat;
        Int tmp1 = idob;

        If ((tmp != 0) & (tmp1 != 0))
        {
            If (messagebox.Show("Изменить запись?", "Изменить запись? ", mes-
sageboxbuttons.YesNo, messageboxicon.Stop) == DialogResult.Yes)
            {
                If ((comboBox1.SelectedValue != null) & (com-
boBox2.SelectedValue != null))
                {
                    Yst_datt.updatequery(Convert.ToInt32(comboBox1.SelectedValue), Con-
vert.ToInt32(comboBox2.SelectedValue), dateTimePicker1.Value, Con-
vert.ToInt32(comboBox3.SelectedValue), Convert.ToInt32(comboBox4.SelectedValue),
checkBox1.Checked, richTextBox1.Text, iddat, idob);
                    Yst_datt.Fill(this.monitoringdataset.yst_dat);
                    ComboBox1.ResetText();
                    ComboBox2.ResetText();
                    ComboBox3.ResetText();
                    ComboBox4.ResetText();
                    CheckBox1.Checked = false;
                    DateTimePicker1.Value = DateTime.Now;
                    RichTextBox1.Clear();
                }
            }
            Else
            {
                MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
            }
        }
    }
    Else
    {
        MessageBox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (NullReferenceException) { MessageBox.Show("Выберите запись для из-
менения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void delbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = iddat;
        Int tmp1 = idob;
        If ((tmp != 0) & (tmp1 != 0))
        {
            If (messagebox.Show("Удалить запись?", "Удалить запись? ", mes-
sageboxbuttons.YesNo, messageboxicon.Stop) == DialogResult.Yes)
            {
                Yst_datt.deletequery(iddat, idob);
                Yst_datt.Fill(this.monitoringdataset.yst_dat);
                ComboBox1.ResetText();
                ComboBox2.ResetText();
                ComboBox3.ResetText();
            }
        }
    }
}

```

```

        Combobox4.resettext();
        Checkbox1.Checked = false;
        Datetimepicker1.Value = datetime.Now;
        Richtextbox1.Clear();
    }
}
Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
}
Catch (nullreferenceexception) { messagebox.Show("Выберите запись для уда-
ления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}
Private void kan_Click(object sender, eventargs e)
{
    K k = new k();
    K.Show();
}
}
}
Yst_el.cs

```

```

Using System;
Using System.Collections.Generic;
Using System.componentmodel;
Using System.Data;
Using System.Drawing;
Using System.Linq;
Using System.Text;
Using System.Windows.Forms;
Using monitoring.monitoringdatasettableadapters;

```

Namespace monitoring

```

{
    Public partial class yst_el : Form
    {
        Public yst_el()
        {
            Initializecomponent();
        }

        Private yst_eltableadapter yst_elt = new yst_eltableadapter();
        Private eltableadapter elt = new eltableadapter();
        Private dattableadapter datt = new dattableadapter();
        Private int iddat = 0;
        Private int idel = 0;
        Private bool status=false;

        Private void yst_el_Load(object sender, eventargs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.y_el". При необходимости она может быть перемещена или удалена.
            This.y_eltableadapter.Fill(this.monitoringdataset.y_el);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.el". При необходимости она может быть перемещена или удалена.
            This.eltableadapter.Fill(this.monitoringdataset.el);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "monitoringdataset.dat". При необходимости она может быть перемещена или удалена.
            This.dattableadapter.Fill(this.monitoringdataset.dat);

            //datagridview1.Height = this.Height - panel1.Height + 5;
            Datagridview1.Height = this.Height - panel1.Height - 35;
        }
    }
}

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		130

```

        Panel2.Width = this.Width - panel3.Width - 10;

        Combobox1.resettext();
        Combobox2.resettext();
        Checkbox1.Checked = false;
        Datetimepicker1.Value = datetime.Now;
        Richtextbox1.Clear();
    }

    Private void yst_el_Resize(object sender, eventargs e)
    {
        //datagridview1.Height = this.Height - panel1.Height + 5;
        Datagridview1.Height = this.Height - panel1.Height - 35;
        Panel2.Width = this.Width - panel3.Width - 10;
    }

    Private void datagridview1_cellclick(object sender, datagridviewcelleventargs
e)
    {
        Try
        {
            Iddat = Con-
vert.toint32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[0].Val
ue);
            Idel = Con-
vert.toint32(this.datagridview1.Rows[this.datagridview1.currentrow.Index].Cells[1].Val
ue);
            If ((iddat > 0) & ((idel > 0)))
            {
                Combobox1.selectedvalue = Con-
vert.toint32(yst_elt.getdatabyid(iddat,idel).Rows[0][0]);
                Combobox2.selectedvalue = Con-
vert.toint32(yst_elt.getdatabyid(iddat, idel).Rows[0][1]);
                Datetimepicker1.Value = Con-
vert.todatetime(yst_elt.getdatabyid(iddat, idel).Rows[0][2]);
                Checkbox1.Checked = Convert.toboolean(yst_elt.getdatabyid(iddat,
idel).Rows[0][3]);
                Richtextbox1.Text = yst_elt.getdatabyid(iddat,
idel).Rows[0][4].toString();
            }
            Else
            {
                Combobox1.resettext();
                Combobox2.resettext();
                Checkbox1.Checked = false;
                Datetimepicker1.Value = datetime.Now;
                Richtextbox1.Clear();
            }
        }
        Catch (nullreferenceexception) { };
    }

    Private void addbutton_Click(object sender, eventargs e)
    {
        If ((combobox1.selectedvalue != null) & (combobox2.selectedvalue != null))
        {
            Yst_elt.insertquery(Convert.toint32(combobox1.selectedvalue), Con-
vert.toint32(combobox2.selectedvalue), datetimepicker1.Value, checkbox1.Checked, rich-
textbox1.Text);
            Yst_elt.Fill(this.monitoringdataset.yst_el);
            Combobox1.resettext();

```

					ПГУ 1.090301.04.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		131

```

        Combobox2.resettext();
        Checkbox1.Checked = false;
        Datetimestpicker1.Value = datetime.Now;
        Richtextbox1.Clear();
    }
    Else
    {
        MessageBox.Show("Есть незаполненные поля", "Внимание!", messageboxbut-
tons.OK, messageboxicon.Warning);
    }
}

Private void edbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = iddat;
        Int tmp1 = idel;

        If ((tmp != 0) & (tmp1 != 0))
        {
            If (messagebox.Show("Изменить запись '" +
elt.getdatabyid(tmp1).Rows[0][1].ToString() + "' в '" +
datt.getdatabyid(tmp).Rows[0][1].ToString() + "' ?", "Изменить запись? ", messagebox-
buttons.yesno, messageboxicon.Stop) == DialogResult.Yes)
            {
                If ((combobox1.selectedvalue != null) & (com-
bobox2.selectedvalue != null))
                {
                    Yst_elt.updatequery(Convert.ToInt32(combobox1.selectedvalue), Con-
vert.ToInt32(combobox2.selectedvalue), datetimestpicker1.Value, checkbox1.Checked, rich-
textbox1.Text, iddat, idel);
                    Yst_elt.Fill(this.monitoringdataset.yst_el);
                    Combobox1.resettext();
                    Combobox2.resettext();
                    Checkbox1.Checked = false;
                    Datetimestpicker1.Value = datetime.Now;
                    Richtextbox1.Clear();
                }
            }
            Else
            {
                MessageBox.Show("Есть незаполненные поля", "Внимание!",
messageboxbuttons.OK, messageboxicon.Warning);
            }
        }
    }
    Else
    {
        Else messagebox.Show("Выберите запись для изменения", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (NullReferenceException) { messagebox.Show("Выберите запись для из-
менения", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void delbutton_Click(object sender, EventArgs e)
{
    Try
    {
        Int tmp = iddat;
        Int tmp1 = idel;

        If ((tmp != 0) & (tmp1 != 0))

```

```

        {
            If (messagebox.Show("Удалить запись?", "Удалить запись? ", mes-
sageboxbuttons.yesno, messageboxicon.Stop) == dialogresult.Yes)
            {
                Yst_elt.deletequery(iddat,idel);
                Yst_elt.Fill(this.monitoringdataset.yst_el);
                Combobox1.resettext();
                Combobox2.resettext();
                Checkbox1.Checked = false;
                Datetimestructure1.Value = datetime.Now;
                Richtextbox1.Clear();
            }
        }
        Else messagebox.Show("Выберите запись для удаления", "Ошибка!", mes-
sageboxbuttons.OK, messageboxicon.Error);
    }
    Catch (nullreferenceexception) { messagebox.Show("Выберите запись для уда-
ления", "Ошибка!", messageboxbuttons.OK, messageboxicon.Error); }
}

Private void checkbox1_checkedchanged(object sender, eventargs e)
{
    Status = checkbox1.Checked;
}

}
}

```

ПРИЛОЖЕНИЕ Г
(Ведомость дипломного проекта)

					ПГУ 1.090301.04.001 ПЗ	Лист
						134
Изм.	Лист	№ докум.	Подпись	Дата		

Формат	Обозначение	Наименование	Кол. лист.
		Документация текстовая	
A4	ПГУ 1.090301.04.001 ПЗ	Пояснительная записка.	112
		Документация графическая	
A4	ПГУ 1.090301.04.002 Э1	Структурная схема системы контроля газового состава.	1
A4	ПГУ 1.090301.04.003 Д2	Контекстная диаграмма функциональной модели.	1
A4	ПГУ 1.090301.04.004 Д2	Диаграмма декомпозиции контекстной диаграммы.	1
A4	ПГУ 1.090301.04.005 Д2	Диаграмма декомпозиции процесса «Учет и анализ показаний датчиков».	1
A4	ПГУ 1.090301.04.006 Д2	Диаграмма декомпозиции «Учет и анализ действий пользователя»	1
A4	ПГУ 1.090301.04.007 Д3	ER- диаграмма логического уровня.	1
A4	ПГУ 1.090301.04.008 Д3	ER- диаграмма физического уровня.	1
A4	ПГУ 1.090301.04.009 Д5	Диаграмма взаимодействия программ	1

					ПГУ 1.090301.04.001 ВД							
Изм.	Лист	№ докум.	Подпись	Дата	Программное обеспечение АРМ оператора системы контроля газо- вого состава Ведомость ВКР			Лит.	Лист	Листов		
Разраб.		Бычинин М.А.								135	135	
Провер.		Трокоз Д.А.						ФВТ гр. 13ВВ1				
Н. Контр.		Кучин А.В.										
Утв.		Пашенко Д.В.										