Федеральное государственное
автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное
управление и проектирование»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.В. Ченцов
«_____»____2018 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.04 – Управление в технических системах

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ВЫБОРА АППАРАТУРЫ ПОЛЕВОГО УРОВНЯ ПРИ ПРОЕКТИРОВАНИИ АСУ ТП

Руководитель _____«___» 062018 г.доц., канд. техн. наук Е.Е. Носкова

Выпускник _____ «___» 062018 г. Д.А.Аблаев

Нормоконтролер _____ «___»062018 г. Т.А. Грудинова

Красноярск 2018

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программное средство для выбора аппаратуры полевого уровня при проектировании АСУ ТП» содержит 95 страницы текстового документа, 19 иллюстрации, 26использованных источников, 1 приложение.

АСУ ТП, ПРОЕКТИРОВАНИЕ, АППАРАТУРА ПОЛЕВОГО УРОВНЯ, ДАТАШИТ,И/ИЛИ ДЕРЕВО, БАЗА ДАННЫХ, SQLЗАПРОС, ВЫБОР АППАРАТУРЫ, ПРИЛОЖЕНИЕ WINDOWSFORMS.

Цель работы: Разработка программного решения, обеспечивающего автоматизацию начального этапа выбора аппаратуры полевого уровня при проектировании АСУ ТП.

В связи с указанной целью были поставлены следующие задачи:

1) анализ обобщенных структур, описывающих всю номенклатуру средств автоматизации, используемых при разработке АСУ ТП;
2) анализ функциональных возможностей и разработка структуры программного решения с учетом потребностей инженеров-проектировщиков АСУ ТП полевого уровня;
3) программная реализация инструментального средства для выбора аппаратуры полевого уровня.

В результате выпускной квалификационной работы было разработан программное решение как инструментальное средство, сопровождающее процесс выбора аппаратуры полевого уровняпри проектировании АСУ ТП.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Процесс проектирования АСУ ТП полевого уровня на современном этапе остается практически не формализованным из-за сложности целей, задач и исходных данный для проектирования. Поэтому задача формализации проектных работ при разработке АСУ ТП остается актуальной и требует решения. Одним из путей решения является формирование исходных данных для проектирования с использованием обобщенных структур на базе И/ИЛИ-деревьев, которые будут описывать всю номенклатуру средств автоматизации (СА), используемых при разработке АСУ ТП.

Актуальность выбранной темы выпускной квалификационной работы связана с программной поддержкой выбора средств автоматизации АСУ ТП.

Таким образом, целью ВКР является разработка программного решения, обеспечивающего автоматизацию начального этапа выбора аппаратуры полевого уровня при проектировании АСУ ТП.

В связи с указанной целью были поставлены следующие задачи:

1) анализ обобщенных структур, описывающих всю номенклатуру средств автоматизации, используемых при разработке АСУ ТП;
2) анализ функциональных возможностей и разработка структуры программного решения с учетом потребностей инженеров-проектировщиков АСУ ТП полевого уровня;
3) программная реализация инструментального средства для выбора аппаратуры полевого уровня.

1 Выбор аппаратуры полевого уровня в процессе проектирования автоматизированных систем управления технологическими процессами

1.1 Этап функционального проектирования АСУ ТП на полевом уровне

Проектирование технического объекта — создание, преобразование и представление в принятой форме образа этого еще не существующего объекта. Образ объекта или его составных частей может создаваться в воображении человека в результате творческого процесса или генерироваться в соответствии с некоторыми алгоритмами в процессе взаимодействия человека и компьютера. В любом случае инженерное проектирование начинается при наличии выраженной потребности общества в некоторых технических объектах, которыми могут быть объекты строительства, промышленные изделия или процессы. Проектирование включает в себя разработку технического предложения и (или) технического задания (ТЗ), отражающих эти потребности, и реализацию ТЗ в виде проектной документации.

Обычно ТЗ представляют в виде некоторых документов, и оно является исходным (первичным) описанием объекта. Результатом проектирования, как правило, служит полный комплект документации, содержащий достаточные сведения для изготовления объекта в заданных условиях. Эта документация и есть проект, точнее окончательное описание объекта. Более коротко, проектирование — процесс, заключающийся в получении и преобразовании исходного описания объекта в окончательное описание на основе выполнения комплекса работ исследовательского, расчетного и конструкторского характера.

Преобразование исходного описания в окончательное порождает ряд промежуточных описаний, подводящих итоги решения некоторых задач и используемых для обсуждения и принятия решений для окончания или продолжения проектирования. Такие промежуточные описания называют проектными решениями.

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека и компьютера, называют автоматизированным проектированием, в отличие от ручного или автоматического (без участия человека на промежуточных этапах). Система, реализующая автоматизированное проектирование, представляет собой систему автоматизированного проектирования (САПР, в англоязычном написании CAD System — Computer Aided Design System).

Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно несложных объектов. Превалирующим в настоящее время является автоматизированное проектирование.

Проектирование сложных объектов основано на применении идей и принципов, изложенных в ряде теорий и подходов. Наиболее общим подходом является системный подход, идеями которого пронизаны различные методики проектирования сложных систем. [1]

Для специалиста в области системотехники идеи и принципы системного подхода являются очевидными и естественными, однако их соблюдение и реализация зачастую сопряжены с определенными трудностями, обусловливаемыми особенностями проектирования. Как и большинство взрослых образованных людей, правильно использующих родной язык без привлечения правил грамматики, инженеры используют системный подход без обращения к пособиям по системному анализу. Однако интуитивный подход без применения правил системного анализа может оказаться недостаточным для решения все более усложняющихся задач инженерной деятельности. [2]

Стадии проектирования — наиболее крупные части проектирования, как процесса, развивающегося во времени. В общем случае выделяют стадии научно-исследовательских работ (НИР), эскизного проекта или опытно-конструкторских работ (ОКР), технического, рабочего проектов, испытаний опытных образцов или опытных партий. Стадию НИР иногда называют предпроектными исследованиями или стадией технического предложения. Очевидно, что по мере перехода от стадии к стадии степень подробности и тщательность проработки проекта возрастают, и рабочий проект уже должен быть вполне достаточным для изготовления опытных или серийных образцов. Близким к определению стадии, но менее четко оговоренным понятием, является понятие этапа проектирования. Проектирование на начальных этапах, в процессе которого принимаются принципиальные проектные решения по облику и принципам действия проектируемых устройств и систем, называют концептуальным проектированием.

Стадии (этапы) проектирования подразделяют на составные части, называемые проектными процедурами. Примерами проектных процедур могут служить подготовка деталировочных чертежей, анализ кинематики, моделирование переходного процесса, оптимизация параметров и другие проектные задачи. В свою очередь, проектные процедуры можно расчленить на более мелкие компоненты, называемые проектными операциями, например, при анализе прочности детали сеточными методами операциями могут быть построение сетки, выбор или расчет внешних воздействий, собственно моделирование полей напряжений и деформаций, представление результатов моделирования в графической и текстовой формах. Проектирование сводится к выполнению некоторых последовательностей проектных процедур — маршрутов проектирования.

Стремление сократить временные затраты на проектирование привело к разработке методик параллельного проектирования (совмещенного проектирования), при котором параллельно во времени решаются задачи, связанные друг с другом по входным и выходным данным таким образом, что

для решения одной из них требуется знание результатов решения другой задачи. Поскольку эти результаты к началу процедуры параллельного проектирования еще не получены, в методике параллельного проектирования должны быть указаны способы задания еще не определенных значений параметров. Примерам параллельного проектирования могут служить параллельная разработка аппаратных и программных средств вычислительных систем или одновременная разработка самолета и средств его аэродромного обслуживания.

Иногда разработку технического задания на проектирование называют внешним проектированием, а реализацию ТЗ — внутренним проектированием.

В ТЗ на проектирование объекта указывают, по крайней мере, следующие данные:

1) назначение объекта;

2) условия эксплуатации. Наряду с качественными характеристиками (представленными в вербальной форме) имеются числовые параметры, называемые внешними параметрами, для которых указаны области допустимых значений. Примеры внешних параметров: температура окружающей среды, внешние силы, электрические напряжения, нагрузки и т.п.;

3) требования к выходным параметрам, т.е. к величинам, характеризующим свойства объекта, интересующие потребителя. [3]

При использовании блочно-иерархического подхода к проектированию представления о проектируемой системе расчленяют на иерархические уровни. На верхнем уровне используют наименее детализированное представление, отражающее только самые общие черты и особенности проектируемой системы. На следующих уровнях степень подробности описания возрастает, при этом рассматривают уже отдельные блоки системы, но с учетом воздействий на каждый из них его соседей. Такой подход позволяет на каждом иерархическом уровне формулировать задачи приемлемой сложности, поддающиеся решению с помощью имеющихся средств проектирования. Разбиение на уровни должно быть таким, чтобы документация на блок любого уровня была обозрима и воспринимаема одним человеком.[4]

В один уровень, как правило включаются представления, имеющие общую физическую основу и допускающие для своего описания использование одного и того же математического аппарата. Когда уровни проектирования выделяют по степени подробности, с которой они отражают свойства проектируемого объекта, то имеют место так называемые горизонтальные уровни проектирования.

Горизонтальные уровни характеризуются следующим:

1) при переходе некоторого уровня, на котором рассматривается система S на соседний, более низкий уровень происходит разделение системы на блоки и расмотрение вместо системы S ее отдельных блоков.

2) рассмотрение каждого из блоков на уровне происходит с большей степенью детализации, чем на уровне.

3) степень восприятия блоков на каждом уровне приблизительна одинакова и достаточна для принятия решения с помощью имеющихся на каждом уровне средств.

4) использование на каждом уровне своих понятий системы и элемента.[5]

Другими словами, блочно-иерархический подход есть декомпозиционный подход (его можно назвать также диакоптическим), который основан на разбиении сложной задачи большой размерности на последовательно и (или) параллельно решаемые группы задач малой размерности, что существенно сокращает требования к используемым вычислительным ресурсам или время решения задач.

Можно говорить не только об иерархических уровнях спецификаций, но и об иерархических уровнях проектирования, понимая под каждым из них совокупность спецификаций некоторого иерархического уровня совместно с постановками задач, методами получения описаний и решения возникающих проектных задач.

Список иерархических уровней в каждом приложении может быть специфичным, но для большинства приложений характерно следующее наиболее крупное выделение уровней:

1) системный уровень, на котором решают наиболее общие задачи проектирования систем, машин и процессов; результаты проектирования представляют в виде структурных схем, генеральных планов, схем размещения оборудования, диаграмм потоков данных и т.п.;

2) макроуровень, на котором проектируют отдельные устройства, узлы машин и приборов, где результаты представляют в виде функциональных, принципиальных и кинематических схем, сборочных чертежей и т.п.;

3) микроуровень, на котором проектируют отдельные детали и элементы машин и приборов.

В каждом приложении число выделяемых уровней и их наименования могут быть различными. Так, в радиоэлектронике микроуровень часто называют компонентным, макроуровень — схемотехническим уровнем. Между схемотехническим и системным уровнями вводят уровень, называемый функционально-логическим уровнем. В вычислительной технике системный уровень подразделяют на уровни проектирования вычислительных систем и вычислительных сетей. В машиностроении имеются уровни деталей, узлов, машин, комплексов.

В зависимости от последовательности решения задач иерархических уровней различают нисходящее, восходящее и смешанное проектирование (стили проектирования). Последовательность решения задач от нижних уровней к верхним характеризует восходящее проектирование, обратная последовательность приводит к нисходящему проектированию, в смешанном стиле имеются элементы как восходящего, так и нисходящего проектирования. В большинстве случаев для сложных систем предпочитают нисходящее

проектирование. Отметим, однако, что при наличии заранее спроектированных составных блоков (устройств) можно говорить о смешанном проектировании.

Неопределенность и нечеткость исходных данных при нисходящем проектировании (так как еще не спроектированы компоненты) или исходных требований при восходящем проектировании (поскольку ТЗ имеется на всю систему, а не на ее части) обусловливают необходимость прогнозирования недостающих данных с последующим их уточнением, т.е. последовательного приближения к окончательному решению (итерационность проектирования).

Наряду с декомпозицией описаний на иерархические уровни применяют разделение представлений о проектируемых объектах на аспекты.

Аспект описания (страта) — описание системы или ее части с некоторой оговоренной точки зрения, определяемой функциональными, физическими или иного типа отношениями между свойствами и элементами.

Различают аспекты функциональный, информационный, структурный и поведенческий (процессный). Функциональное описание относят к функциям системы и чаще всего представляют его функциональными схемами. Получение функциональных описаний часто называют функциональным проектированием.Информационное описание включает в себя основные понятия предметной области (сущности), словесное пояснение или числовые значения характеристик (атрибутов) используемых объектов, а также описание связей между этими понятиями и характеристиками. Информационные модели можно представлять графически (графы, диаграммы сущность-отношение), в виде таблиц или списков. Получение информационных описаний часто называют информационным проектированием или применительно к созданию баз данных — инфологическим проектированием.Структурное описание относится к морфологии системы, характеризует составные части системы и их межсоединения и может быть представлено структурными схемами, а также различного рода конструкторской документацией. Получение конструкторской документации, т.е. описание геометрических форм изделий, состава компонентов и их пространственного размещения, называют конструкторским проектированием.

Поведенческое описание характеризует процессы функционирования (алгоритмы) системы и (или) технологические процессы создания системы. Разработка алгоритмов и программного обеспечения систем является предметом алгоритмического проектирования, а разработка технологических процессов изготовления изделий — предметом технологического проектирования.

Иногда аспекты описаний связывают с подсистемами, функционирование которых основано на различных физических процессах.

В общем случае выделение страт может быть неоднозначным. Так, помимо указанного подхода. очевидна целесообразность выделения таких аспектов, как функциональное (разработка принципов действия, структурных, функциональных, принципиальных схем), конструкторское (определение форм

и пространственного расположения компонентов изделий), алгоритмическое (разработка алгоритмов и программного обеспечения) и технологическое (разработка технологических процессов) проектирование систем. Примерами страт в случае САПР могут служить также рассматриваемые далее виды обеспечения автоматизированного проектирования.[6]

Автоматизированная система управления технологическим процессом – это человеко-машинная система управления, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления технологическим объектом в соответствии с принятым критерием. Технологический объект управления (ТОУ) – это совокупность технологического оборудования и реализованного в нем по соответствующим инструкциям или регламентам технологического процесса производства. Такое определение АСУТП подчеркивает наличие в её составе современных автоматических средств сбора и обработки информации, в первую очередь вычислительной техники ; роль человека, в системе как субъекта труда, принимающего содержательное участие в выработке решений по управлению; реализацию в системе процесса обработки технологической и технико-экономической информации; цель функционирования АСУТП, заключающаяся в оптимизации работы технологического объекта управления по принятому критерию (критериям) управления путем соответствующего выбора управляющих воздействий. Критерий управления АСУТП – это соотношение, характеризующее качество функционирования технологического объекта управления в целом и принимающее конкретные числовые значения в зависимости от используемых управляющих воздействий. Критерием управления обычно является технико-экономический показатель (например, себестоимость выходного продукта при заданном его качестве, производительность ТОУ при заданном качестве выходного продукта и т.п.) или технический показатель (например, параметры процесса, характеристики выходного продукта). АСУТП осуществляет управление в целом в момент протекания технологического процесса и в выработке и реализации решений по управлению с участием средств вычислительной техники и человека-оператора.

В управлении технологическим процессом можно выделить три фазы:

1) получение и первичная обработка информации;
2) анализ полученной информации и принятие решений;
3) реализация управляющих воздействий.

Для современных АСУТП характерно стремление автоматизировать все три фазы управления и свести к минимуму участие оператора- технолога в управлении процессом, возложив на него задачу контроля за процессом и работой автоматизированной системы в случае её отказа. Для успешного контроля за ходом процесса и эффективного выполнения функций резервирования оперативному персоналу необходима разнообразная информация.

Функции АСУТП подразделяются на информационные, управляющие и вспомогательные. Информационная функция АСУТП – это функция системы, содержанием которой является сбор, обработка и представление информации о состоянии технологического процесса оперативному персоналу или передача этой информации для последующей обработки. Управляющая функция АСУТП – это функция, результатом которой является выработка и реализация управляющих воздействий на технологический процесс. К управляющим функциям АСУТП относятся:

1) формирование и передача на исполнительные устройства управляющих сигналов;

2) определение рационального режима технологического процесса;

3) выдача оператору рекомендаций по управлению технологическим процессом.

Вспомогательные функции АСУТП – это функции, обеспечивающие решение внутренних задач. Вспомогательные функции системы предназначены, прежде всего, для обеспечения собственного функционирования АСУТП (обеспечение заданного алгоритма функционирования технических средств системы, контроль их состояния, хранение информации и т.п.). Основные функции, реализуемые в АСУТП, следующие:

Контроль технологических параметров, а также состояний вспомогательного оборудования (механизмом и приводных электродвигателей, режимов работы регуляторов, положение запорной арматуры, процент открытия регулирующих органов и т.п.) и отображение оператору в виде текущих цифровых значений в единицах параметра или трендов(графиков), регистрация параметров на рабочих станциях оператора и последующая передача их на сервер с заданной дискретностью, визуальная сигнализация и текстовое сообщение оператору, а также мигающая световая сигнализация с одновременной подачей звукового сигнала об отклонениях оперативных технологических параметров от их нормальных значений, а также неисправностях в системе управления.

На рисунке 1 представлена схема комплекса технических средств многоуровневой системы управления, обобщающая многочисленные применения таких систем для управления технологическими процессами нефтяной и газовой промышленности.

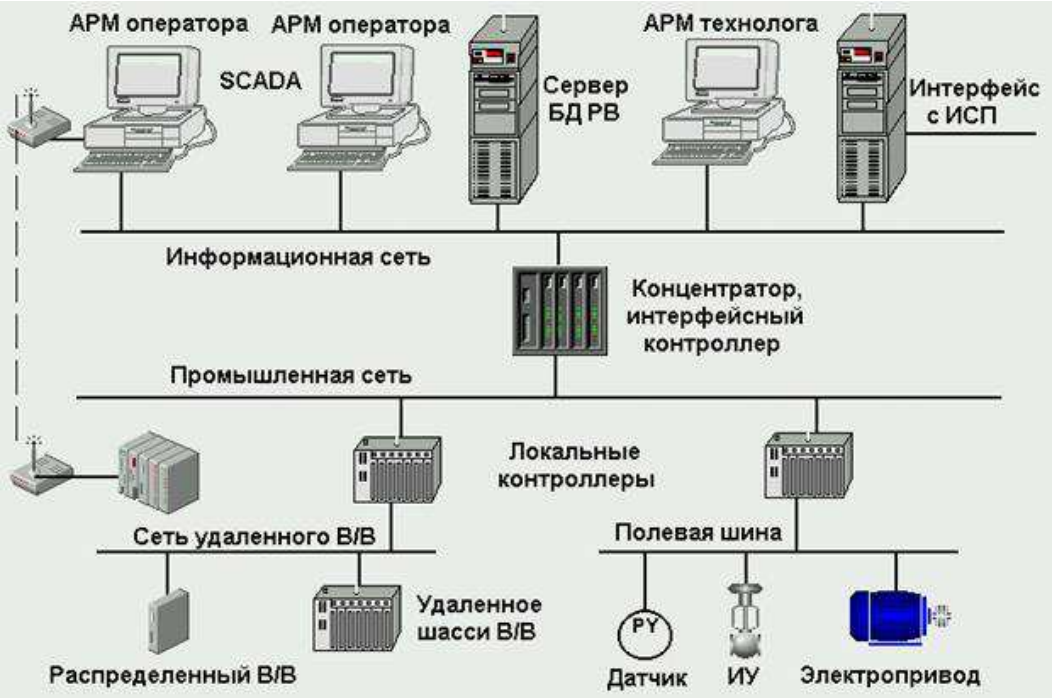


Рисунок 1 – Комплекс технических средств АСУ ТП

Как правило, это двух- или трехуровневые системы, и именно на этих уровнях реализуется непосредственное управление технологическими процессами. Специфика каждой конкретной системы управления определяется используемой на каждом уровне программно - аппаратной платформой.

Структура АСУ ТП на полевом уровне в качестве элементов содержит различные датчики (измерительные преобразователи) для сбора информации о ходе технологического процесса, электроприводы и исполнительные устройства для реализации регулирующих и управляющих воздействий.

Датчики поставляют информацию локальным контроллерам (PLC), которые могут обеспечить реализацию следующих функций:

1) сбор, первичная обработка и хранение информации о состоянии;
2) оборудования и параметрах технологического процесса;
3) автоматическое логическое управление и регулирование;
4) исполнение команд с пункта управления;
5) самодиагностика работы программного обеспечения и состояния самого контроллера;
6) обмен информацией с пунктами управления.

Таким образом результатом проектирования АСУ ТП является проектная документация на систему автоматизации, которая содержит графические материалы: структурные схемы контроля и управления; функциональные схемы, автоматизации; принципиальные электрические и пневматические схемы; схемы внешних соединений, чертежи общих видов щитов и пультов; монтажные схемы щитов и пультов. Эта документация обеспечивает возможность заказа технических средств автоматизации, осуществление

монтажных и наладочных работ, эксплуатацию систем автоматизации, а также оценкустоимости технических средств, их монтажа.

Задание на проектирование систем автоматизации технологических процессов содержит следующие данные:

1) основание для проектирования;

2) наименование предприятия и перечень автоматизируемых отделений, цехов, аппаратов и агрегатов;

3) этапы проектирования;

4) технологические схемы производства по отдельным видам продукции с перечней применяемого оборудования, а также коммуникаций для продукта, теплоносителя и т. д.; номенклатуру контролируемых и регулируемых технологических параметров;

5) чертежи отделений и цехов с указанием места расположения технологического оборудования и трубных коммуникаций, а также щитов и пультов управления; чертежи автоматизируемого технологического оборудования с конкретизацией мест установки технических средств автоматизации;

6) строительные чертежи отделений и цехов;

7) схемы энергоснабжения, силового питания и управления электрическими двигателями;

8) схемы водоснабжения и воздухоснабжения;

9) особые условия (взрывоопасность, пожароопасность помещений и т. д.).

Автоматизация проектирования такого сложного объекта как АСУ ТП на современном этапе частично осуществлена на горизонтальных и вертикальных уровнях процесса проектирования. В основном решаются вопросы автоматизации этапа конструкторского проектирования. На этапе функционального проектирования зачастую нет ни коммерческих, ни отраслевых решений по автоматизации функционального проектирования АСУ ТП. Поэтому локальные решения задач автоматизации функционального проектирования АСУ ТП на разных уровнях является актуальными.

1.2 Задача выбора аппаратуры полевого уровня при реализации проектной процедуры синтеза

Структурный синтез - наиболее трудная для формализации проектнаяпроцедура. В существующих САПР в большинстве случаевсинтез выполняет человек, а компьютер используется для верификациипредлагаемых вариантов. Дальнейшее повышение степени автоматизациипроектных работ определяется в первую очередь успехамив постановке и алгоритмизации структурного синтеза. Процедурыструктурного синтеза классифицируются по ряду признаков.По целям синтеза и содержанию получаемыхрезультатов выделяют следующие процедуры структурного синтеза:

1) выбор принципов построения и функционирования технических объектов;

2) выбор технического решения;

3) синтез технической документации.

Формулировка целей структурного синтеза зависитпрежде всего от стадии проектирования.

Выбор принципов построения и функционирования техническихобъектов производится на стадиях предпроектных исследований инаучно-исследовательских работ. Его цель - установление физических,информационных, организационных принципов и т. п. В машиностроениитакую задачу часто называют определением обликатехнического объекта. При проектировании посредством компьютера содержанием этойпроцедуры являются выбор архитектурных решений и построениеструктурных схем.

Выбор технического решения производится преимущественно настадиях опытно-конструкторских работ в рамках ранее установленныхпринципов функционирования и имеет целью получение функциональных,принципиальных, кинематических схем, конструктивныхрешений, технологических маршрутов изготовления деталейи т. п.

Синтез технической документации относится к стадиям техническогои рабочего проектирования и заключается в автоматическомпреобразовании данных о схемах и конструкциях, выраженныхна внутреннем языке САПР, в текстовую и чертежную документацию,оформленную по правилам Единой системы конструкторскойдокументации (ЕСКД).

По трудностям формализации процедур синтеза выделяют пять уровней сложности.

Задачи первого уровня сложности появляются там, где структурапроектируемого объекта предопределена результатами ранеевыполненных исследований и синтез сводится к выбору числовыхзначений параметров для заданной структуры. Задачи первогоуровня — это задачи параметрического синтеза.

Задачи второго уровня сложности заключаются в выборе структурыиз конечного множества вариантов при условиях:

1) все вариантызаранее известны либо их можно легко получить;

2) мощностьмножества вариантов настолько мала, что возможен полныйперебор при их сравнительной оценке.

Задачи третьего уровня сложности также сводятся к выборуварианта в конечном множестве, но мощность множества достаточновелика, чтобы реализовать полный перебор.

Задачи четвертого уровня сложности характеризуются выборомварианта структуры в множествах, мощность которых априорно неизвестнаи не исключена возможность, что она неограничена.

Задачи пятого уровня сложности связаны с поиском решений,основанных на новых, ранее неизвестных или неиспользовавшихсяидеях и принципах. В задачах предыдущих уровней существованиерешений не подвергалось сомнению и требовалось найти наилучшееили приемлемое решение. В задачах пятого уровня достижениерешения равноценно получению принципиально нового типа техническихобъектов.

По типу синтезируемыхструктур различают процедурыодномерного, схемного и геометрического синтеза.

Одномерный синтез заключается в построении одномерных последовательностейиз элементов некоторой природы. Примеры такихпоследовательностей - описания технологических процессов вформе маршрутной или операционной технологии, вычислительныхпроцессов - в виде алгоритмов и программ длякомпьютера.

Схемный синтез связан с разработкой различных схем - функциональных,структурных, кинематических, принципиальных и т. п.,отражающих результаты проектирования объектов до конкретизациих геометрических форм.

Геометрический синтез выполняется при конструировании изделийи связан с определением их геометрических форм (синтез формы)и с расположением объекта или его частей в пространствеотносительно заданных ориентиров (задачи позиционирования).

Формализация сведений об объектах синтеза. Сведения обобъектах синтеза могут включать в себя данные о базовых элементах(примитивах), макроэлементах, законченных и обобщенныхструктурах проектируемых объектов. Под объектами синтеза понимаютсяпредставления о реальных объектах, выражаемые в видеописаний проектируемых изделий и процессов на одном из языковпроектирования, т. е. чертежи, схемы, списки, тексты и т. п.

Базовый элемент - элементарная часть синтезируемогообъекта, которую невозможно или нецелесообразно разделять наболее мелкие составные части. При синтезе механизмов в роли примитивоввыступают детали, при синтезе радиоэлектронных схем - электрорадиоэлементы(резисторы, конденсаторы, трансформаторыи т. п.), при проектировании функциональных схем– логическиеэлементы (дизъюнкторы, конъюнкторы, триггеры и т. п.).Основную часть сведений об элементе составляет перечень егосвойств, называемых атрибутами. В перечне указываются именаатрибутов, в сведения могут также входить значения атрибутов,ссылки на другие элементы и процедуры, с

помощью которых конкретизируютсясвойства или выявляются допустимые типы связейи т. п.

Макроэлемент - типовая совокупность взаимосвязанныхбазовых элементов, используемая при синтезе наравне с базовымиэлементами. Примеры макроэлементов - макроопределения припрограммировании, типовые графические изображения (зубчатыхколес, подшипников, транзисторов) о при синтезе конструкторскойдокументации, сборочные единицы при конструировании. Для макроэлементов,как и для базовых элементов, задаются сведения обатрибутах и дополнительно о структуре макроэлементов.

Законченная структура - совокупность взаимосвязанныхпримитивов и макроэлементов, представляющая собой возможныйвариант структуры проектируемого объекта. В практике проектированияшироко используются типовые проекты, конструкции исхемы. Примеры законченных структур при проектировании локальных вычислительных сетей - структуры сетей Ethernet,Cambridge ring и др., при проектировании суперкомпьютер– структурыCray-1, Cyber-205, STARANидр.

Обобщенная структура строится на основе многих частныхзаконченных структур. В ней отражаются все элементы, макроэлементыи связи между ними, которые встречались хотя бы в однойиз частных структур. Одинаковые части входят в обобщеннуюструктуру без дублирования, что делает ее более лаконичной посравнению с набором описаний типовых проектов.

В перечисленных формах сведений об объектах можно выделитьописания собственно элементов и отношений между ними. С помощьюотношений элементы объединяются в макроэлементы иструктуры. Отношения между элементами могут быть следующихтипов: классификации («целое - часть», «род - вид»), упорядоченияво времени (временные) и в пространстве (пространственные),признаковые и взаимодействия посредством связей информационных,электрических, кинематических, оптических и т. п.

В СИИ для формализации декларативных знаний применяютспособы, основанные на понятиях фрейма и семантической сети.

Фрейм - структура данных, в которой в определенном порядкепредставлены сведения о свойствах объекта. Типичный видфрейма(имя фрейма;$a_1 = p_1;\ a_2 = p_2; \ldots;\ a_n = p_n;\ q_1, q_2, \ldots, q_m$),
где a_i - имя i-го атрибута, p_i - его значение, q_i - ссылка на некоторыйдругой фрейм или процедуру.

С помощью фреймов можно описывать иерархические структуры.При этом в качестве a_i иp_i в (3.1) используются имя и значениеi-го слота. Слотом называется вложенная во фрейм структураданных, т. е. слот по отношению к фрейму - иерархически подчиненнаяинформационная единица. Организация слота аналогичнаструктуре (3.1). В общем случае во фрейме может быть более

двухиерархических уровней.Фреймы - естественная форма представления сведений об элементах синтезируемых объектов в системах структурного синтеза.В случае макроэлементов сведения об их структуре помещаютсяво фрейм верхнего уровня, а атрибуты и их значения для примитивов,составляющих макроэлемент, включены в слоты.Различают символические и конкретные фреймы. В символическихфреймах, называемых фреймами- прототипами, атрибутыявляются переменными, для них указывается только область определения.Пример символического фрейма: <резистор; номинал=
Xl; мощность=X2; класс точности=X3; тип конструкции=X4; ГОСТ=X5>, где X1 ... X5 - переменные, которые могутпринимать значения из соответствующих множеств.

В конкретных фреймах атрибуты принимают фиксированныезначения, например: <АРМ; тип=АРМ2-05; заводской номер=37; основная ЭВМ= СМ1407; операционная система=ОС РВ; назначение=АРМ конструктора; структура предприятия;вычислительная сеть САПР>, где «структура предприятия» и«вычислительная сеть САПР» - ссылки на другие фреймы (слоты).Совокупность конкретных фреймов в некоторой подсистеме САПРпредставляет собой базу данных. Функции СИИ выполняются наоснове символических фреймов с помощью операций поиска по образцу,наполнения слотов данными, введения новых фреймов - прототипови т. п. При выполнении операции поиска по образцусимволический фрейм с частично заполненными слотами входит взапрос, ответом на который будет один или несколько конкретныхфреймов из базы данных, не противоречащих данным, содержащимсяв запросе.

Семантическая сеть - форма представления знаний в видесовокупности понятий и отношений между ними внекоторойпредметной области. Семантическую сеть удобно изображать графом,в котором вершины отображают понятия, а ребра – отношениямежду ними. Семантические сети можно использовать дляпредставления структур проектируемых объектов, если вершинампоставить в соответствие элементы, а ребрам - соединения элементовили другие взаимодействия. В качестве вершин могут фигурироватьфреймы, представляющие сведения об элементах.

В системах структурного синтеза семантические сети используютсядля представления обобщенных структур проектируемыхобъектов. Обобщенные структуры имеют иерархическую организацию,отражают множество вариантов построения объектов и называютсяИ/ИЛИ - деревьями.

В И/ИЛИ - дереве вершины делятся на ярусы (уровни). Вершинымогут быть двух типов И и ИЛИ. Вершины И отображаютварианты технической реализации составных частей объектов,а вершины или - составные части

объектов, выделенные по функциональномупризнаку. Если на верхнем ярусе вершины или соответствуютфункциональному назначению систем из некоторогокласса, то вершины И соседнего яруса отображают способы реализациисистем. Вершины ИЛИ и И последующих ярусов отображаютфункции и способы построения подсистем, блоков, узлов и базовыхэлементов. Ветви в И/ИЛИ - дереве представляют классификационныеи признаковые отношения между понятиями. Каждой вершинеИ в И/ИЛИ - дереве может быть поставлен в соответствиенекоторый предикат, характеризующий условия выбора этой вершины(альтернативы) при решении конкретных задач синтеза.

На рисунке2, а приведен пример И/ИЛИ - дерева для классатехнических объектов, называемых вторичными источниками электропитания(ВИП). Вершины или показаны прямоугольниками,вершины И - овалами. Сплошной линией изображены ветви, соответствующиеклассификационным отношениям, пунктирной – признаковымотношениям. [7]

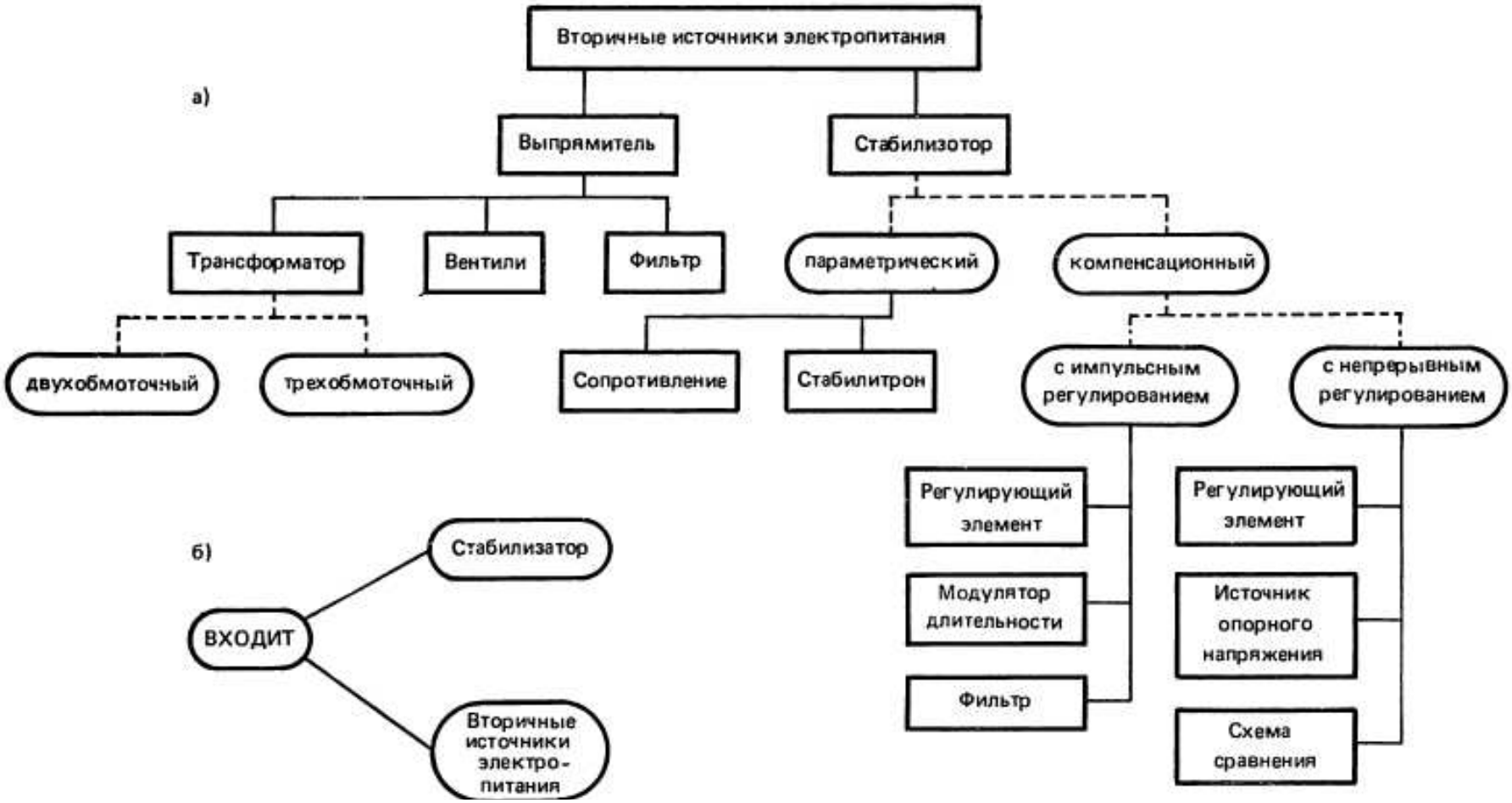


Рисунок 2 – Семантические сети в виде И/ИЛИ – дерева для вторичных источников питания (а) и фрагмент представления в виде двудольного графа (б)

Одним из походов к формализации и автоматизации проектных процедур параметрического и структурного синтеза является метод морфологического анализа, суть которого заключается в построении морфологического И/ИЛИ дерева, определяющего избыточную структуру АСУ ТП, в которой представлены все возможные варианты наборов средств автоматизации. Сужение пространства поиска оптимальной структуры АСУ ТП достигается за счет применения морфологических таблиц, построенных на основе классификационных признаков СА. Для расширения номенклатуры рассматриваемых проектировщиком СА необходимо перейти от ручных методов проектирования к автоматизированным, которые позволяют получить и просмотреть большое количество проектных решений при разработке структуры АСУ ТП.

Разрабатываемая методика проектирования базируется на формализации процедуры параметрического и структурного синтеза на основе метода морфологического анализа и включает представление АСУ ТП на трех уровнях:

1) уровень идентификации, где определяются классификационные признаки необходимого средства автоматизации для разработки функционирования АСУТП;

2) уровень спецификации, где составляются списки функций обследуемых классов СА и на их основе разрабатываются морфологические таблицы возможных альтернатив по каждой функции;

3) уровень реализации и интеграции, где разрабатывается модель предметной области, которая представляет собой модель класса устройств, дополненную алгоритмами синтеза этих устройств по техническому заданию.

Трехуровневое представление АСУ ТП, изображенное на рисунке 3, есть совокупность моделей, включающая как избыточную обобщенную информационную структуру АСУ ТП, так и структуру АСУ ТП полевого уровня, удовлетворяющую требованиям ТЗ на проектирование (рис. 1).

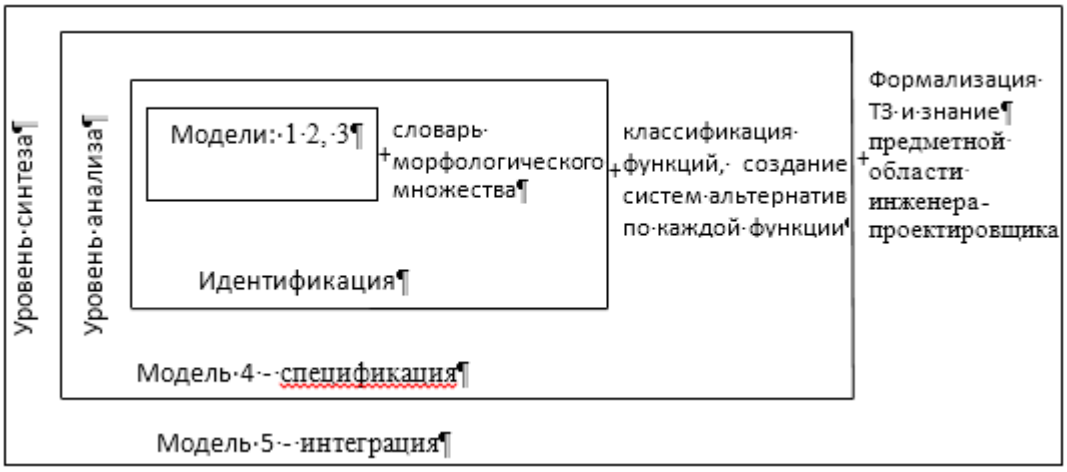


Рисунок3 - Структура разработанной методики в виде трехуровневой интегративной модели

На уровне идентификации выделяются классификационные признаки и сводятся в систему, если присутствует несколько классов, то определяется отношение между классами, тем самым получается упорядоченное множество, представленное как взаимодействие трех моделей 1-3.

Для создания модели 1 необходимо провести морфологический анализ класса основных физических величин рабочих сред, которые измеряются, контролируются или регулируются СА, а так же факторов, влияющих на функционирование первичного средства автоматизации. Для создания модели 2 необходимо провести морфологический анализ классов СА. Модель 2 разрабатывается в виде морфологического И/ИЛИ-дерева классификации первичного СА. Модель 3 разрабатывается в виде морфологического И/ИЛИ-дерева множества типов оборудования и позволяет получить характеристики оборудования, которые следует учитывать при выборе СА, предназначенного для установки на конкретном месте технологического объекта управления.

Модель 4 позволяет провести всесторонний анализ любого устройства принадлежащего рассматриваемому классу. Разработанные морфологические таблицы для модели 4 позволяют провести анализ, не упустив из виду ни одного параметра, при выборе СА и рассмотреть все возможные варианты исполнения выбираемого СА, представленные на рынке. Модель 5 разрабатывается в виде морфологического И/ИЛИ дерево множества типов оборудования технологического объекта управления и является моделью 4, дополненной знаниями, необходимыми для синтеза объектов данного класса.

Рассмотрим модель 1, которая представлена на рисунке 4 виде морфологического И/ИЛИ-дерева множества функций первичного средства автоматизации.Данное морфологическое И/ИЛИ дерево множества функций первичного средства автоматизации оптимизирует практические знания инженера - проектировщика. Его использование уменьшает вероятность появления ошибки при определении, к какому классу принадлежит первичное средство автоматизации.

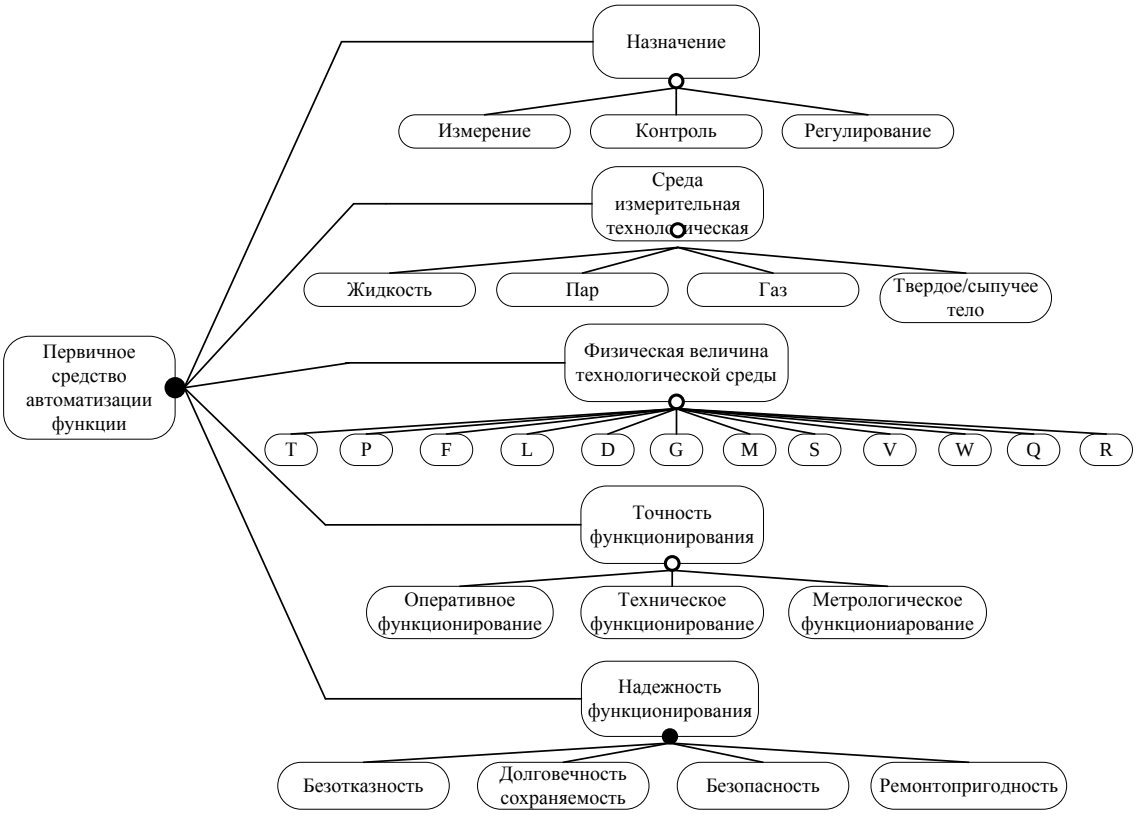


Рисунок4 - Морфологическое И/ИЛИ дерево множества функций первичного средства автоматизации

В качестве исходных данных после анализа современного полевого оборудования, используемого при проектировании АСУ ТП, были составлены И/ИЛИ деревья для всех существующих средств автоматизации, которые будут использоваться при автоматизации проектных работ. На рисунке 5 представлен пример обобщенной структуры в виде И/ИЛИдерева для такого типа СА, как расходомеры.

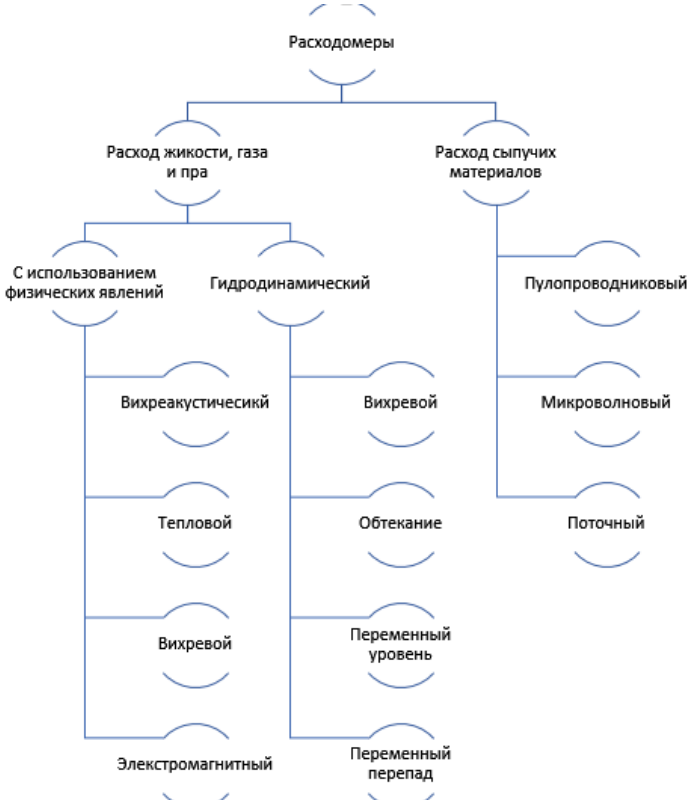


Рисунок5 - Представление обобщенной структуры в виде И/ИЛИ дерева

На рисунках 6 и 7 представлены обобщенные структуры в виде И/ИЛИ деревьев для таких СА как:

- термометры;
- уровнемеры.

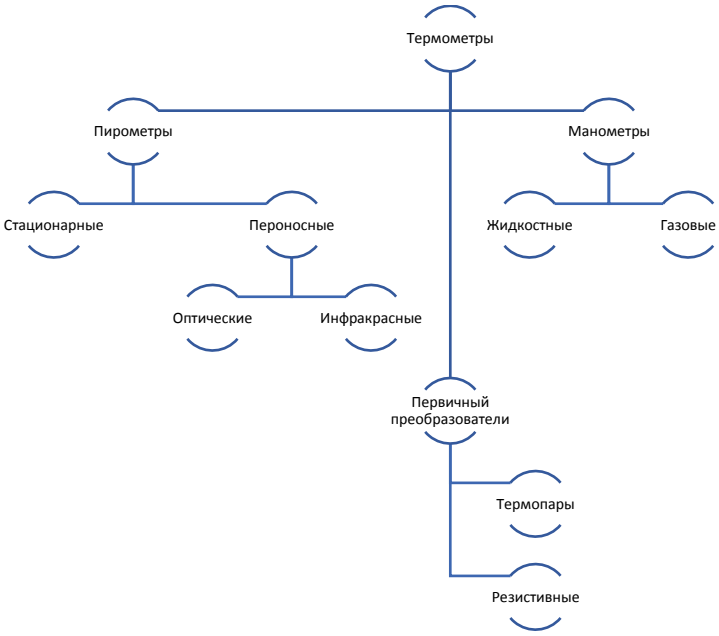


Рисунок 6 - Представление обобщенной структуры в виде И/ИЛИ дерева

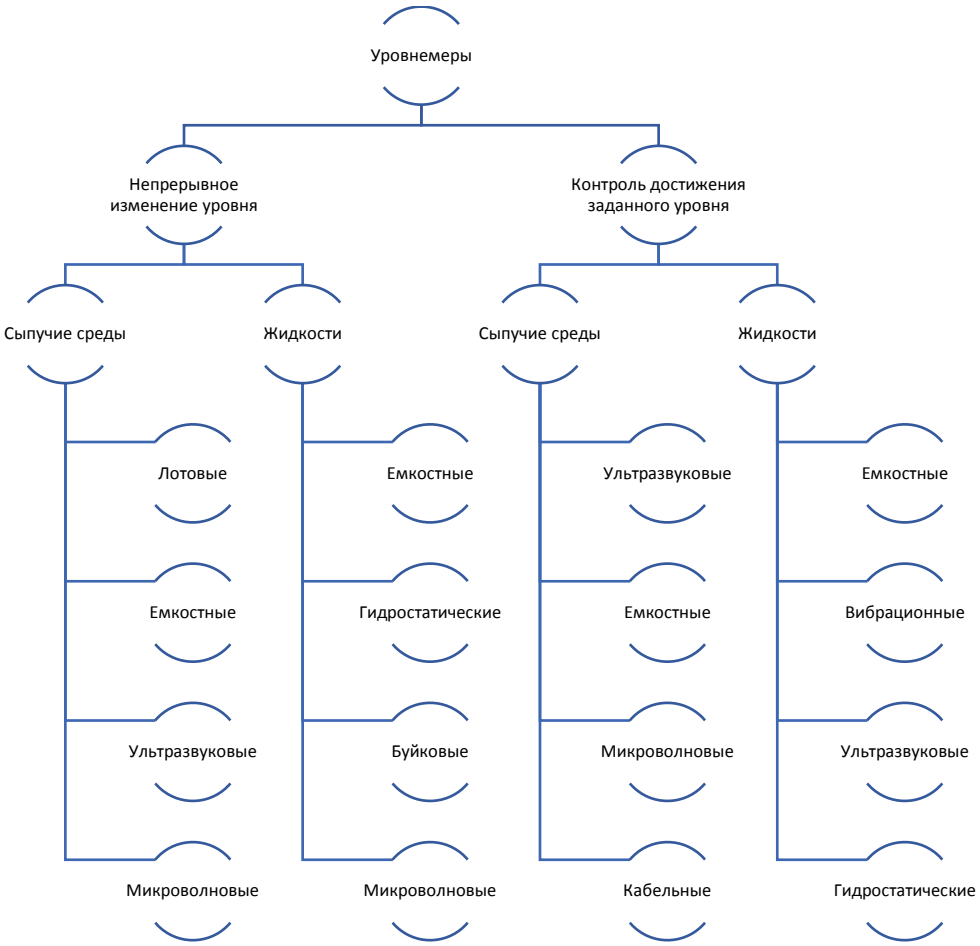


Рисунок 7 - Представление обобщенной структуры в виде И/ИЛИ дерева

Аналогичные деревья составлены и реализованы для всех существующих типов СА. Для того чтобы получить по И/ИЛИдереву решение задачи синтеза, нужно выполнить следующую последовательность шагов [1, 2]:

1) выбрать одну из связок, исходящих из корневой вершины.

2) для каждого потомка корневой вершины выбрать по одной исходящей связке.

3) продолжать этот процесс до тех пор, пока каждая из полученных вершин-потомков не станет концевой вершиной в альтернативном дереве.[8]

2 Программные средства для решения задачи выбора аппаратуры полевого уровня

2.1 Инструментальные средства автоматизация проектирования АСУ ТП полевого уровня

На рынке современных коммерческих САПР решений, которые могут быть использованы в качестве инструмента проектирования АСУ ТП полевого уровня на этапе функционального проектирования для решения задач синтеза нет. Зачастую САПР решают отдельные задачи функционального проектирования и комплекс задач конструкторского проектирования. Рассмотрим известные для проектировщиков АСУ ТП инструментальные средства.

EPLAN предлагает платформу для сквозного проектирования, охватывающего следующие отрасли:электротехника,КИПиА,гидравлика/пневматикаимеханика(проектирование шкафов и жгутов). Благодаря открытой архитектуре и стандартным интеграционным модулям EPLAN может быть экономически эффективно интегрирован с большим спектром сторонних решений: системами механического проектирования,ERPиPDMсистемами, системами проектирования зданий, промышленных производств и кораблей.

Широко применяется в следующих отраслях:

- автомобилестроение;
- машиностроение;
- металлургия;
- химическая и фармацевтическая промышленность;
- пищевая промышленность;
- добыча нефти и газа;
- трубопроводный транспорт;
- нефте- и газо-переработка;
- производство тепла и электроэнергии;
- передача и распределение электроэнергии;
- железнодорожный транспорт;
- водоснабжение и водоотведение;
- станкостроение;
- легкая промышленность;
- автоматизация зданий.

Основные модули платформы:

– EPLAN Electric P8— модульное и масштабируемое решение для электротехнического проектирования, автоматического создания проектной и рабочей документации;

– EPLANFluid- программное обеспечение для проектирования пневмо/гидроавтоматики, систем смазки и охлаждения и автоматического создания соответствующей проектной и рабочей документации;

– EPLANProPanel - 3D проектирование электротехнических шкафов с передачей данных в производство. Виртуальное трехмерное моделирование, создание двух- и трехмерных чертежей, трехмерное изображение проводных и маршрутных схем, наличие шаблонов для работы сверлильного оборудования и интеграция со станками ЧПУ;

– EPLANFieldSys- работа с планами трасс;

– EPLANHarnessProD– создание документации для производства жгутовых соединений;

– EPLANPrePlanning- программное обеспечение для предварительного (эскизного) проектирования объектов и генерации проектной документации;

– EPLANEngineeringConfigurator— автоматическое создание схем и моделей шкафов;

– EPLANCogineer- решение для автоматического создания электрических и пневмогидравлических схем;

– EPLANSyngineer- новая коммуникационная и информационная платформа облегчает мехатронные операции между всеми участниками процесса проектирования;

– EPLANSmartWiring -визуализирует компоновку оборудования в шкафу, монтируемые устройства и их соединения, а также кабельные трассы, основываясь на соответствующих моделях из EPLAN Pro Panel.

CADdy - программное обеспечение, использующееся для2D и 3D проектирования.Решения на базе CaDdyиспользуются для автоматизированного производства (CAM), электротехники, метода конечных элементов (FEM), ГИС (графическая информационная технология) и управления данными о продуктах (PDM).

Также существуют модули, предназначенные для:

– архитектуры;
– электротехники;
– инжиниринга;
– стальные конструкций;
– обследования;
– планирование событий;
– электроустановок;
– конструирования корпусов.

3D-модули также доступны, но старая версия CADdy (CADdy classic) основана на технологии DOS, поэтому функциональность 3D и ориентация объектов довольно старомодны. Кроме того, не соответствует обычным программам Windows.

AutoCAD — двух- и трёхмерная система автоматизированного проектирования и черчения, разработанная компанией Autodesk. Первая версия системы была выпущена в 1982 году. AutoCAD и специализированные приложения на его основе нашли широкое применение в машиностроении, строительстве, архитектуре и других отраслях промышленности. Программа выпускается на 18 языках. Уровень локализации варьирует от полной адаптации до перевода только справочной документации. Русскоязычная версия локализована полностью, включая интерфейс командной строки и всю документацию, кроме руководства по программированию.

Ранние версии AutoCAD оперировали небольшим числом элементарных объектов, такими как круги, линии, дуги и текст, из которых составлялись более сложные. В этом качестве AutoCAD заслужил репутацию «электронного кульмана», которая остаётся за ним и поныне. Однако на современном этапе возможности AutoCAD весьма широки и намного превосходят возможности «электронного кульмана».

В области двумерного проектирования AutoCAD по-прежнему позволяет использовать элементарные графические примитивы для получения более сложных объектов. Кроме того, программа предоставляет весьма обширные возможности работы со слоями и аннотативными объектами (размерами, текстом, обозначениями). Использование механизма внешних ссылок (XRef) позволяет разбивать чертёж на составные файлы, за которые ответственны различные разработчики, а динамические блоки расширяют возможности автоматизации 2D-проектирования обычным пользователем без использования программирования. Начиная с версии 2010 в AutoCAD реализована поддержка двумерного параметрического черчения. В версии 2014 появилась возможность динамической связи чертежа с реальными картографическими данными (GeoLocation API).

Широкое распространение AutoCAD в мире обусловлено не в последнюю очередь развитыми средствами разработки и адаптации, которые позволяют настроить систему под нужды конкретных пользователей и значительно расширить функциональность базовой системы. Большой набор инструментальных средств для разработки приложений делает базовую версию AutoCAD универсальной платформой для разработки приложений. На базе AutoCAD самой компанией Autodesk и сторонними производителями создано большое количество специализированных прикладных приложений, таких как:

- AutoCAD Mechanical;
- AutoCAD Electrical;
- AutoCAD Architecture;
- GeoniCS;
- Promis-e;

– PLANT-4D;
– AutoPLANT;
– СПДС GraphiCS;
– MechaniCS;
– GEOBRIDGE;
– САПРЛЭП;
– Rubius Electric Suite идругих.

Ряд отечественных решений представлен на платформе AutoCad, большинство из которых являются корпоративными решениями, не имеющими коммерческого рынка сбыта.

2.2 Использование MSDN при разработке программных средств для решения задачи выбора аппаратуры полевого уровня при проектировании АСУ ТП

Выбор элементов АСУ ТП проектировщику, т.е. получение проектного решения желательно проводить за приемлемое время с использованием эргономичного инструментария. Поэтому ставится задача разработки программного средства, с помощью которого осуществляется информационное сопровождение процесса определения элементов АСУ ТП полевого уровня, которые удовлетворяют ТЗ на проектирование. На рисунке 8 представлена совокупность программных модулей программного решения.

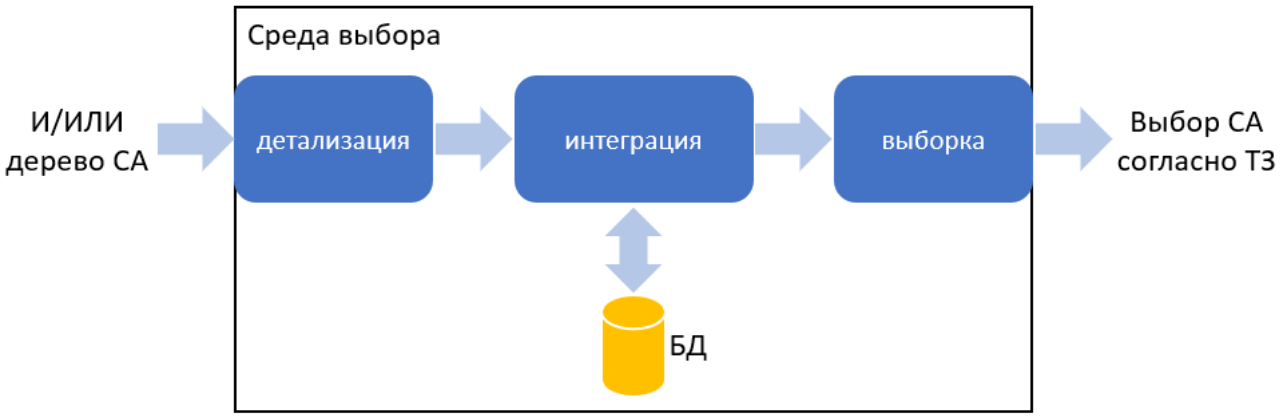


Рисунок8 – Совокупность программных модулей

Для реализации структуры, представленной на рисунке 8 используются современные компоненты и средства программирования.

MicrosoftDeveloperNetwork (MSDN) — подразделение компании Майкрософт, ответственное за взаимодействие фирмы с разработчиками. В данном случае, под разработчиками понимаются разработчики аппаратного

обеспечения, интересующиеся операционной системой, а также разработчики, использующие программные интерфейсы операционной системы и скриптовые языки различных приложений, разработанных Microsoft. Такое взаимодействие с разработчиками имеет несколько форм: веб-сайты, новостные рассылки, конференции разработчиков, блоги, рассылка CD/DVD. Жизненный цикл взаимодействия с разработчиками варьируется от поддержки уже устаревших продуктов до распространения информации о новых возможностях.

Подразделение работает как информационный сервис для разработчиков программного обеспечения. Основное внимание (в последнее время) уделяется платформе Microsoft .NET, но присутствуют и статьи, охватывающие такие области как практика программирования и шаблоны проектирования. Многие ресурсы доступны бесплатно в онлайн, другие — только по почте на основе платной подписки. В зависимости от уровня подписки, подписчики могут получать ранние редакции новых версий операционных систем или других продуктов Майкрософт (приложения MicrosoftOffice, VisualStudio и др.). Некоторые университеты включены в программу MSDNAcademicAlliance (MSDNAA), что позволяет студентам получать преимущества подписки MSDN.

Библиотека MSDN (MSDN Library) — библиотека официальной технической документации для разработчиков под ОС Microsoft Windows. MSDN. Библиотека MSDN содержит документацию на API продуктов Microsoft, а также код примеров, технические статьи и другую полезную для разработчиков информацию. Она бесплатно доступна через Интернет, а также на CD и DVD для подписчиков MSDN. Изначально версия на диске была доступна только как часть подписки MSDN и распространялась каждый квартал. Однако, с 2006 года ISO-образ может быть бесплатно скачан с сайта Microsoft, а диски стали выходить не каждый квартал, а вместе с крупными релизами (Visual Studio или Windows, а также с пакетами обновлений).

Visual Studio Express интегрируется только с MSDN Express Library, которая является ограниченной версией полной библиотеки. Тем не менее, полная версия также может быть бесплатно скачана с сайта Microsoft и отдельно установлена.

Каждая версия библиотеки MSDN может быть доступна только с одного обозревателя справки (Microsoft Document Explorer или иного), который встроен в текущую версию Visual Studio (или иногда в две версии). Каждая новая версия Visual Studio не интегрируется с более старой версией библиотеки MSDN, но совместимая версия библиотеки MSDN выходит вместе с каждым новым релизом Visual Studio и поставляется на диске вместе с ним. Новые версии библиотеки MSDN также не интегрируются с более старыми версиями Visual Studio, а также не включают документацию на устаревшие функции или

снятые с поддержки продукты. Несколько версий библиотеки MSDN могут быть одновременно установлены и функционировать на одной машине.

При создании программного решения были использованы следующие классы и объекты среды разработки MicrosoftVisualStudio 2017 Community,программной платформы .NETFramework и языка программирования C#.

Класс SQLConnection

Класс представляет из себя подключение к базе данных SQL Server. Система подключения клиента к серверу базы данных является эквивалентом сетевого подключения к серверу. SqlConnection используется совместно с SqlDataAdapter и SqlCommand для повышения производительности при подключении к базе данных Microsoft SQL Server.

Конструкторы

SQLConnection(String) - Инициализирует новый экземпляр класса SqlConnection после получения строки подключения.

Свойства

ConnectionString - Получает или задает строку, используемую для создания подключения к базе данных SQL Server.

Методы

– Open() - Открывает подключение к базе данных со значениями свойств, определяемыми объектом ConnectionString.

– Close() - Закрывает соединение с базой данных. Рекомендуется использовать этот метод для закрытия любого открытого подключения.

Исключения

Если исключение SqlException создается методом и, выполняющим SqlCommand уровень важности не превышает 19, SqlConnection остается открытым. Когда уровень важности превышает 20, сервер обычно закрывает SqlConnection. Тем не менее, пользователь может опять открыть подключение и продолжить работу.

Приложению, которое создает экземпляр объекта SqlConnection могут потребоваться прямые и косвенные вызовы, чтобы иметь достаточных разрешений для установки декларативной или принудительной безопасности. Параметры безопасности SqlConnection устанавливаются при помощи объекта SqlClientPermission. Пользователи могут проверить наличие необходимых разрешений с помощью объекта SqlClientPermissionAttribute. Пользователи и Администраторы могут также использовать Caspol.exe (Code Access Security Policy Tool) для изменения политики безопасности на уровне пользователя, компьютера, и предприятия.[9]

Класс SqlCommandBuilder

SqlCommandBuilder() – необходим для автоматического создания однотабличных инструкций таких как INSERT, UPDATE, DELETE, используемых для согласования изменений, внесенных в DataSet, со связанной базой данных SQLServer.

Для того, чтобы SQLComandBuilder автоматически создал инструкции INSERT, UPDATE или DELETE, необходимо изначально задать свойство SelectCommand для получения требуемого набора данных. Свойство SelectCommand должно возвратить по крайней мере один столбец, содержащий первичный ключ или столбец с атрибутом UNIQUE. Если это не удается, InvalidOperation создается исключение, и команды не создаются. При изменении SelectCommand после получения данных, например, после первого обновления, необходимо вызвать метод RefreshSchema для обновления метаданных.

Конструкторы

SqlCommandBuilder(SqlDataAdapter) - Инициализирует новый экземпляр SqlCommandBuilder класс со связанным объектом SqlDataAdapter.

Методы

– GetInsertCommand() - Возвращает автоматически созданный объект SqlCommand, необходимый для выполнения операций вставки в базу данных.

– GetUpdateCommand() - Возвращает автоматически созданный объект SqlCommand, необходимый для выполнения обновлений в базе данных.

– GetDeleteCommand() - Возвращает автоматически созданный SqlCommand объект, необходимый для выполнения операций удаления в базе данных.[10]

Класс SqlDataAdapter

Представляет набор команд данных и подключение к базе данных, которые используются для заполнения DataSet и обновления базы данных SQL Server.

SqlDataAdapter используется в качестве моста между DataSet и SQL Server для извлечения и сохранения данных. С помощью соответствующих инструкций Transact-SQL SqlDataAdapter использует Fill для сопоставления изменений DataSet в соответствии с теми данными, что находятся источнике, и Update для обратной операции. Обновление выполняется построчно. Для каждой вставленной, измененной и удаленной строки метод Update определяет тип изменения, которое было выполнено.

Конструкторы

– SqlDataAdapter() - Инициализирует новый экземпляр класса SqlDataAdapter.

– SqlDataAdapter(String,SqlConnection) - Инициализирует новый экземпляр SqlDataAdapter, где в качестве начальных свойств задаются класс SelectCommand и объект SqlConnection.

Свойства

– SelectCommand – Возвращает, задает инструкцию Transact-SQL или хранит процедуру, используемую для выбора записей в источнике данных.

– InsertCommand – Возвращает, задает инструкцию Transact-SQL или хранит процедуру для вставки новых записей в источнике данных.

– UpdateCommand – Возвращает, задает инструкцию Transact-SQL или хранит процедуру, используемую для обновления записей в источнике данных.

– DeleteCommand – Возвращает, задает инструкцию Transact-SQL или хранит процедуру для удаления записей из набора данных.

Методы

– Fill(DataTable) - Добавляет или обновляет строки указанного диапазона в DataSet для соответствия строкам в источнике данных с помощью DataTable.

– Update(DataTable) - Обновляет значения в базе данных, выполнив соответствующие инструкции INSERT, UPDATE или DELETE для каждой вставки, обновления или удаления строки в указанном DataTable.

События

Данные события создаются так как происходит попытка обновления базы данных.

– RowUpdated – Событие происходит после обновления данных в источнике.

– RowUpdating - Событие происходит в процессе обновления данных в источнике. Выполняется попытка обновления, поэтому создается событие.[11]

Класс DataTable

Представляет одну таблицу данных в памяти. DataTable является центральным объектом в библиотеке ADO.NET. Другие объекты, такие как DataSet и DataView включают в себя DataTable.

Конструкторы

– DataTable () - Инициализирует новый экземпляр класса DataTable без аргументов.

– DataTable (String) - Инициализирует новый экземпляр класса DataTable с указанным именем таблицы.

Свойства

– DataSet – Возвращает название DataSet, которому принадлежит DataTable.

– Rows - Возвращает коллекцию строк, принадлежащую данному DataTable.

Методы

Select(String) – Возвращает массив всех объектов DataRow, которые были отфильтрованы по определенному критерию.

События

– ColumnChanged - Происходит после того, как значение было изменено для указанной DataColumn в DataRow .

– ColumnChanging - Происходит при изменении значения для указанной DataColumn в DataRow .

– Disposed - Добавляет обработчик событий для прослушивания события Disposed на компоненте.

– Initialized - Происходит после инициализации DataTable .

– RowChanged - Происходит после успешного изменения DataRow .

– RowChanging - Возникает при изменении DataRow .

– RowDeleted - Происходит после удаления строки в таблице.

– RowDeleting - Происходит до того, как строка в таблице будет удалена.

– TableCleared - Происходит после очистки DataTable .

– TableClearing - Происходит при очистке DataTable .

– TableNewRow - Происходит при вставке нового DataRow .[12]

Класс DataGridView

Класс DataGridView отображает данные в виде таблицы

Конструкторы

DataGridView() - Инициализирует новый экземпляр класса DataGridView.

Свойства

– ColumnCount - Получает или задает число столбцов, отображаемых в объекте DataGridView.

– Columns - Возвращает коллекцию, содержащую все столбцы элемента управления.

– CurrentCell - Возвращает или задает ячейку, которая является активной в данный момент.

– CurrentRow - Возвращает строку, содержащую текущую ячейку.

– DataSource - Получает или задает источник данных, для которого объект DataGridView отображает данные.

– Events - Возвращает список обработчиков событий, которые прикреплены к этому Component.

– RowCount - Получает или задает число строк, отображаемых в элементе управления DataGridView.

– Rows - Получает коллекцию, содержащую все строки в элементе управления DataGridView.

– SelectedCells - Возвращает коллекцию ячеек, выбранных пользователем.

– SelectedColumns - Возвращает коллекцию столбцов, выбранных пользователем.

– SelectedRows - Возвращает коллекцию строк, выбранных пользователем.

– SelectionMode - Получает или задает значение, указывающее, каким образом могут быть выбраны ячейки объекта DataGridView.

Методы

– OnClick(EventArgs) - Вызывает событие Click.(Наследуется от Control.)

– OnDoubleClick(EventArgs) - Вызываетсобытие DoubleClick.(Переопределяет Control.OnDoubleClick(EventArgs).)

– OnMouseClick(MouseEventArgs) - Вызывает событие MouseClick.

– OnMouseDoubleClick(MouseEventArgs) - Вызывает событие MouseDoubleClick.

– Update() - Вызывает перерисовку элементом управления недопустимых областей клиентской области.(Наследуется от Control.)

События

– CellMouseDown - Возникает при нажатии кнопки мыши, когда указатель мыши находится в пределах ячейки.

– CellMouseEnter - Возникает при наведении указателя мыши на ячейку.

– CellMouseLeave - Возникает, когда указатель мыши выходит за пределы ячейки.

– CellMouseMove - Происходит, когда указатель мыши перемещается в элемент управления DataGridView.

– CellMouseUp - Возникает, если пользователь отпускает кнопку мыши, когда указатель мыши находится на ячейке.

– CellDoubleClick - Возникает при щелчке в любой части ячейки.[13]

Класс MessageBox

Отображает окно сообщения (диалоговое окно) с текстом для пользователя. Это модальное окно, блокирующее другие действия в приложении, пока пользователь не закроет его. MessageBox может содержать текст, кнопки и символы, с помощью которых информируется и инструктируется пользователь.

Методы

Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String) - Отображает окно сообщения с заданным текстом, заголовком, кнопками, значком, кнопкой по умолчанию, параметрами для выбора, кнопкой "Справка", используя заданный файл справки и Ключевое слово справки.[14]

Класс SqlDataReader

Предоставляет способ чтения последовательного потока строк из базы данных SQL Server.

Свойства

– Connection - Возвращает ключ SqlConnection, ассоциированный с экземпляром SqlDataReader.

– Item[String] - Возвращает значение указанного столбца в собственном формате при наличии заданного имени столбца.

Методы

– Close() - Закрывает объект SqlDataReader.

– Read() - Перемещает SqlDataReader к следующей записи.

– ReadAsync() - Асинхронная версия Read, который перемещает модуль чтения к следующей записи в результирующем наборе.[15]

Класс Form

Представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения. Класс Form является представлением любого окна, отображаемого в приложении. Класс Form может использоваться для создания стандартных форм, плавающих форм и форм, у которых отсутствуют грани. Также можно использовать класс Form для создания модальных окон, таких как диалоговое окно. Можно использовать форму в качестве начального класса в приложении, разместив метод с именем Main в главном классе.

С помощью свойств, доступных в классе Form, можно определить внешний вид, размер, цвет и компоненты управления стандартного или диалоговое окна.

События Form позволяют реагировать на действия, выполняемые в форме. Можно использовать Activated событий для выполнения операций, таких как обновление данных, отображаемых в элементах управления формы, когда форма активируется.

Конструкторы

Form() - Инициализирует новый экземпляр класса Form.

Свойства

– Events - Возвращает список обработчиков событий, которые прикреплены к этому Component.

– Location - Получает или задает объект Point, который представляет собой верхний левый угол формы Form в экранных координатах.

– MaximizeBox - Получает или задает значение, указывающее, отображается ли кнопка Развернуть в строке заголовка формы.

– MinimizeBox - Получает или задает значение, указывающее, отображается ли кнопка Свернуть в строке заголовка формы.

– StartPosition - Возвращает или задает начальное положение формы в режиме выполнения.

Методы

– Close() - Закрывает форму.

– OnLoad(EventArgs) - Вызывает событие Load.

– Show() - Отображает элемент управления.

– ShowDialog(IWin32Window) - Отображает эту форму в виде модального диалогового окна с указанным владельцем.

События

Load - Происходит до первоначального отображения формы.[16]

Класс TreeView

Отображает иерархическую коллекцию помеченных элементов, каждый из которых представлен TreeNode. Коллекция Nodes содержит все объекты TreeNode. Узлы дерева в этой коллекции, называются корневые узлы дерева. Все узлы, добавляемые к корневому узлу дерева, называется дочерними. Узлы дерева можно развернуть, чтобы отобразить следующий уровень дочерних узлов дерева.

Конструкторы

TreeView() - Инициализирует новый экземпляр класса TreeView.

Свойства

– Nodes - Возвращает коллекцию узлов дерева, которая назначена элементу управления иерархического представления.

– SelectedNode - Возвращает или задает узел дерева, который в настоящий момент выбран в элементе управления иерархического представления.

– TopNode - Возвращает или задает первый полностью отображаемый узел дерева в элементе управления иерархического представления.

Методы

– BeginUpdate() - Отключает любую перерисовку представления в виде дерева.

– EndUpdate() - Разрешает перерисовку представления в виде дерева.

– OnMouseClick(MouseEventArgs) - Вызывает событие MouseClick.(Наследуется от Control.)

– Add() – Добавляет новый элемент в дерево.[17]

Класс TreeNode

Представляет узел TreeView. Коллекция Nodes содержит все дочерние объекты TreeNode, назначенные текущему TreeNode. Вы можете добавлять, удалять, копировать узлы. При этом все дерево, дочерние узлы будут добавлены, удалены или скопированы. Каждая TreeNode может содержать коллекцию других TreeNode объектов. Узлам и вершинам можно присваивать названия и изображения. Узлы дерева можно развернуть, чтобы отобразить следующий уровень дочерних узлов дерева.

Конструкторы

TreeNode() - Инициализирует новый экземпляр класса TreeNode.

Свойства

– FirstNode - Возвращает первый дочерний узел дерева в коллекции узлов дерева.

– Index - Возвращает позицию узла дерева в коллекции узлов дерева.

– IsSelected - Возвращает значение, указывающее, является ли узел дерева в выбранном состоянии.

– Name - Возвращает или задает имя узла дерева.

– Nodes - Возвращает коллекцию TreeNode объектов, назначенных текущему узлу дерева.

– TreeView - Возвращает родительский дерево, которому назначен узел дерева.

Методы

BeginEdit() - Инициирует редактирование метки узла дерева.[18]

Структура Point

Представляет упорядоченную пару из целых чисел со знаком x и y координаты, определяющую точку на двумерной плоскости.

Конструкторы

– Point(Int32, Int32) - Инициализирует новый экземпляр Point класса с указанными координатами.

– Point(Size) - Инициализирует новый экземпляр Point класса Size.

Свойства

– IsEmpty - Возвращает значение, указывающее, пуст ли массив Point.

– X - Возвращает или задает координату x Point.

– Y - Возвращает или задает координату y Point.

Методы

Add(Point, Size) - Добавляет указанный Size в указанный Point.

Поля

– Round(PointF) - Преобразует указанный PointF для Point объекта, округляя Point значения до ближайшего целого числа.

– Subtract(Point, Size) - Возвращает результат вычитания указанного Size из указанного Point.

– ToString() - Преобразует этот Point для восприятия строку.(Переопределяет ValueType.ToString().)

– Truncate(PointF) - Преобразует указанный PointF для Point путем усечения значений Point.

Операторы

– Addition(Point, Size) - ПреобразуетPointподанногоSize.

– Equality(Point, Point) - Сравнивает два объекта Point. Результат указывает ли значения X и Y двух Point объекты равны.

– Explicit(Point to Size) - Преобразует указанный Point структуру Size структуры.

– Implicit(Point to PointF) - Преобразует указанный Point структуру PointF структуры.

– Inequality(Point, Point) - Сравнивает два объекта Point. Результат указывает ли значения X или Y двух Point объекты не равны.

– Subtraction(Point, Size) - Преобразует Point при помощи отрицательного значения заданного Size.[19]

Класс OpenFileDialog

Отображает диалоговое окно, позволяющее пользователю открыть файл. Этот класс не наследуется.

Этот класс позволяет проверить, существует ли файл и открыть его. ShowReadOnly Свойство определяет, отображается ли в диалоговом окне флажок только для чтения. ReadOnlyChecked указывает, установлен ли флажок «только для чтения».

Большинство основных функциональных возможностей для этого класса находится в FileDialog класса.

Конструкторы

OpenFileDialog() - Инициализирует экземпляр класса OpenFileDialog.

Свойства

– FileName - Возвращает или задает строку, содержащую имя файла, выбранного в диалоговом окне.

– SafeFileName - Возвращает имя и расширение файла, выбранного в диалоговом окне. Имя файла не включает сведения о пути.

– ShowReadOnly - Получает или задает значение, указывающее, имеется ли в диалоговом окне флажок "доступно только для чтения".

Методы

– OpenFile() - Открывает выбранный пользователем файл в режиме "только чтение". Файл задается свойством FileName.

– ShowDialog() - Запускает общее диалоговое окно с заданным по умолчанию владельцем.

– ShowDialog(IWin32Window) - Запускает общее диалоговое окно с указанным владельцем.

События

FileOk - Происходит, когда пользователь щелкает кнопку Открыть или Сохранить в диалоговом окне файла.[20]

Класс Timer

Создает события, повторяющиеся через заданное время. Также можно задать время первоначальной задержки срабатывания.

Конструкторы

– Timer() - Инициализирует новый экземпляр класса Timer и задает всем свойствам начальные значения.

– Timer(Double) - Инициализирует новый экземпляр класса Timer и задает свойству Interval указанное значение в миллисекундах.

Свойства

– Enabled - Возвращает или задает значение, определяющее, должен ли объект Timer вызывать событие Elapsed.

– Events - Возвращает список обработчиков событий, которые прикреплены к этому Component.(Наследуется от Component.)

– Interval - Возвращает или задает интервал в миллисекундах, по истечении которого возникает событие Elapsed.

Методы

– Close() - Освобождает ресурсы, используемые объектом Timer.

– Start() - Начинает вызывать событие Elapsed, задавая для свойства Enabled значение true.

– Stop() - Прекращает вызывать событие Elapsed, задавая для свойства Enabled значение false.[21]

Класс MenuStrip

MenuStrip – Класс выпадающего меню. MenuStrip является контейнером верхнего уровня, который заменяет MainMenu.

Конструкторы

MenuStrip() - Инициализирует новый экземпляр класса MenuStrip.

Свойства

– Events - Возвращает список обработчиков событий, которые прикреплены к этому Component.

– Handle - Возвращает дескриптор окна, с которым связан элемент управления.

– ImageList - Возвращает или задает список изображений, содержащий изображение, отображаемое на элементе ToolStrip.

– Items - Возвращает все элементы, которые принадлежат к объекту ToolStrip.

– Text - Возвращает или задает текст, связанный с этим элементом управления.

Методы

– Contains(Control) - Возвращает значение, указывающее, является ли указанный элемент управления дочерним элементом.

– CreateControl() - Вызывает принудительное создание видимого элемента управления, включая создание дескриптора и всех видимых дочерних элементов.

События

– Click - Происходит при щелчке элемента управления.

– MouseClick - Вызывается при щелчке мышью элемента управления.[22]

Класс MenuStrip

Класс MenuStrip - представляет контекстное меню, вызываемое нажатием на правую кнопку мыши. ContextMenuStrip заменяет ContextMenu. Для отображения контекстного выпадающего меню по нажатию на правую кнопку мыши его можно связать с любым элементом управления. Также контекстное меню можно показать программно с помощью метода Show.

Контекстные меню обычно используют для объединения различных элементов управления программой, связанных определенным контекстом.

Конструкторы

– ContextMenuStrip() - Инициализирует новый экземпляр класса ContextMenuStrip.

– ContextMenuStrip(IContainer) - Инициализирует новый экземпляр ContextMenuStrip класса и связывает его с указанным контейнером.

Методы

– CreateControl() - Вызывает принудительное создание видимого элемента управления, включая создание дескриптора и всех видимых дочерних элементов. Например, контекстное меню можно использовать для изменения параметров элемента управления TextBox таких как: шрифт, размер шрифта, цвет.

– Close() – Закрывает элемент управления ToolStripDropDown.

События

Click - Происходит при щелчке элемента управления.[23]

Интерфейс Application

Представление приложения Microsoft Excel.[24]

Объект Excelworkbook

Представляет книгу в проектах Excel, созданную с помощью инструментов разработчика Office в VisualStudio.

Методы

– Activate() - Активирует первое окно, связанное с книгой.

– Close(Object, Object, Object) - Закрывает книгу.

– Save() - Сохраняет изменения в книге.

– SaveAs(Object, Object, Object, Object, Object, Object, XlSaveAsAccessMode, Object, Object, Object, Object, Object) - Сохраняетизмененияв книгевдругойфайл.

– SaveCopyAs(Object) - Сохраняет копию книги в файл, но не изменяет открытую книгу в памяти.[25]

Интерфейс Worksheet

Представляет страницы в проектах Excel, созданные с помощью средств разработки Office в VisualStudio.

Методы

– Activate() - Делает текущий лист активным листом.

– Close(Object, Object, Object) - Закрываетк нигу.

– Save() - Сохраняет изменения в книге;

– При реализации программного решения были использованы вышеописанные программные компоненты.[26]

3 Технология разработки программного средства выбора аппаратуры полевого уровня при проектировании АСУ ТП

3.1 Визуальное моделирование программного решения

Для представления взаимодействия объектов программного решения используется UMLдиаграммапоследовательностей, отображающая взаимодействие объектов системы в динамике.На рисунке 9 представлена диаграмма последовательности для программного средства выбора аппаратуры полевого уровня.

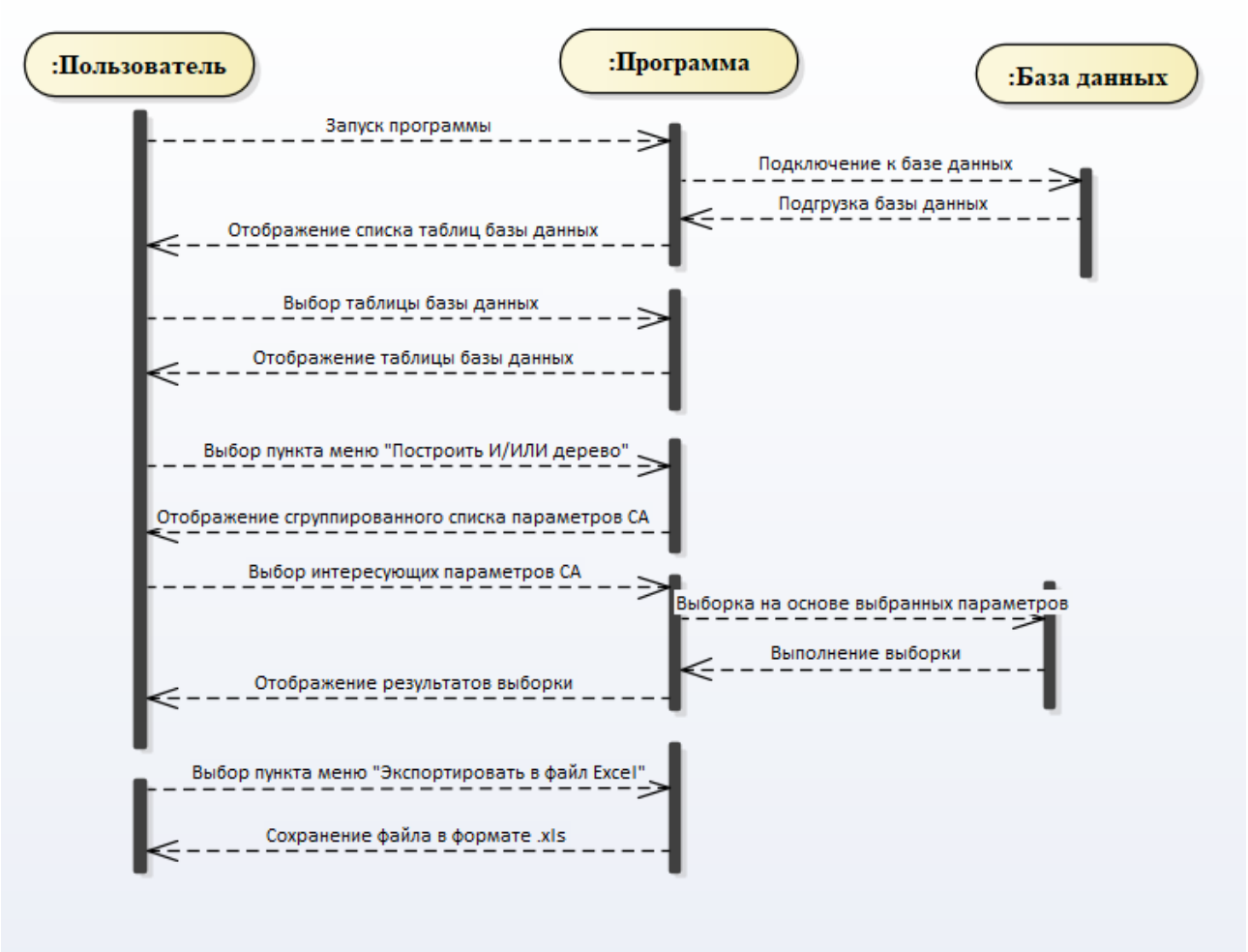


Рисунок 9 – Диаграмма последовательности

Диаграммы последовательностей — это отличное средство документирования поведения системы, детализации логики сценариев использования.Диаграмма последовательностей относится к диаграммам взаимодействия UML, описывающим поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени. Другими словами,

диаграмма последовательностей отображает временные особенности передачи и приема сообщений объектами.

Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними представлена на рисунке 10.

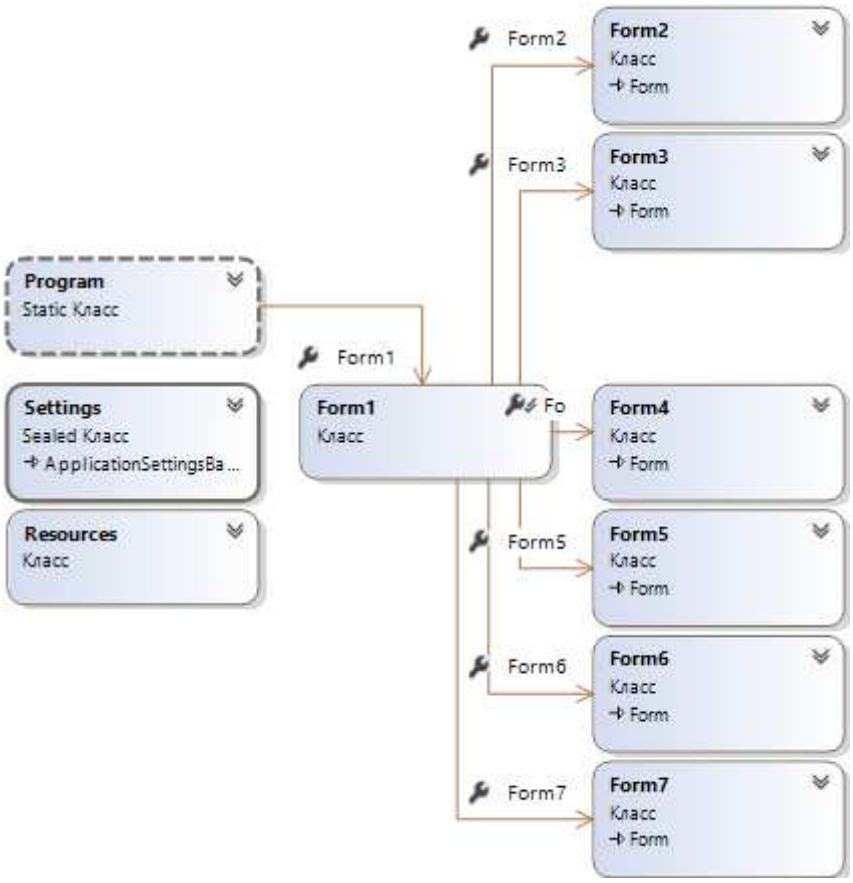


Рисунок 10 – Диаграмма классов

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

1) в верхней части написано имя класса. имя класса выравнивается по центру и пишется полужирным шрифтом. имена классов начинаются с заглавной буквы. если класс абстрактный — то его имя пишется полужирным курсивом;

2) посередине располагаются поля (атрибуты) класса. они выровнены по левому краю и начинаются с маленькой буквы;

3) нижняя часть содержит методы класса. они также выровнены по левому краю и пишутся с маленькой буквы.

3.2 Программная реализация пользовательского интерфейса

Form1

Главная форма программы, которую пользователь видит при запуске изображена на рисунке 11.На данной форме расположены:

а) основное окно в виде DataGridView, куда импортированы данные из таблицы базы данных MSSQL;

б) элемент ContextMenuStrip, расположенный в шапке главной формы, где находятся пункты меню:

- файл;
- команды.

В пункте меню файл находятся подпункты:

- подгрузить базу данных, для подключения локального файла базы данных к программе;
- сохранить базу данных, для подтверждения изменений в определенной таблице или базе данных в целом;
- выход, для завершения работы с программой и закрытия всех открытых окон.

В пункте меню команды находятся подпункты:

- создать запрос SQL, для ручного ввода необходимого SQL запроса;
- добавить строку, для добавления новой строки в текущую открытую таблицу;
- добавить столбец, для добавления нового столбца в текущую открытую таблицу;
- построить И/ИЛИ дерево, для генерации дерева, при помощи которого будет осуществляться выборка интересующих средств автоматизации;
- экспортировать таблицу в Excel файл, для сохранения текущей таблицы в файл Excel.

в) скрывающееся окно “Подключение БД”, в котором расположен древовидный список TreeView, отражающий структуру БД, а также элемент ContextMenuStrip, содержащий кнопки управления:

- загрузить базу данных;
- изменить макет данных, для изменения структуры выбранной таблицы;
- обновить список таблиц.

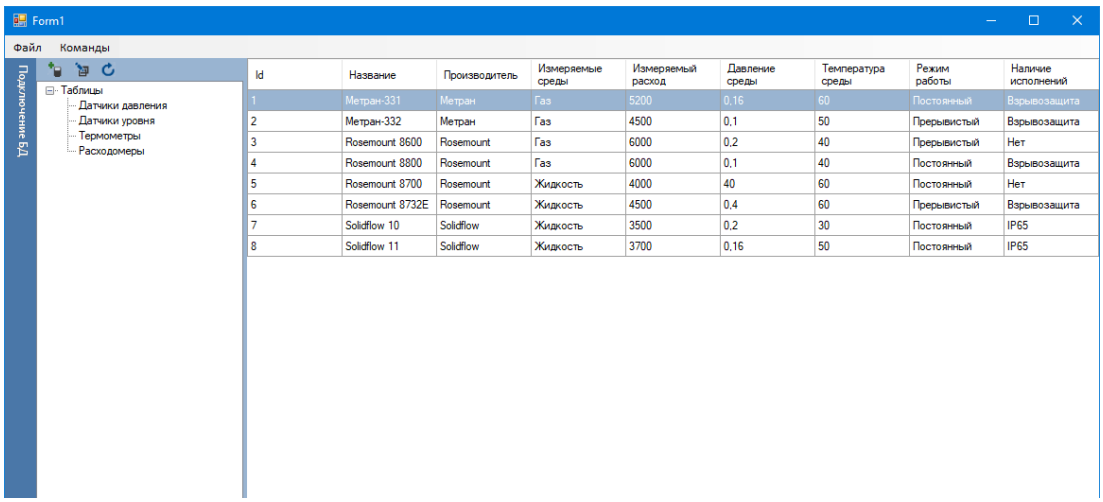


Рисунок11 - Главная форма программы

Form2

Форма, генерирующая И/ИЛИ дерево на основе группированной выборки средств автоматизации изображена на рисунке 12. На данной формепосредством нажатия на соответствующие элементы дерева происходит выборка СА на основе выбранных параметров.

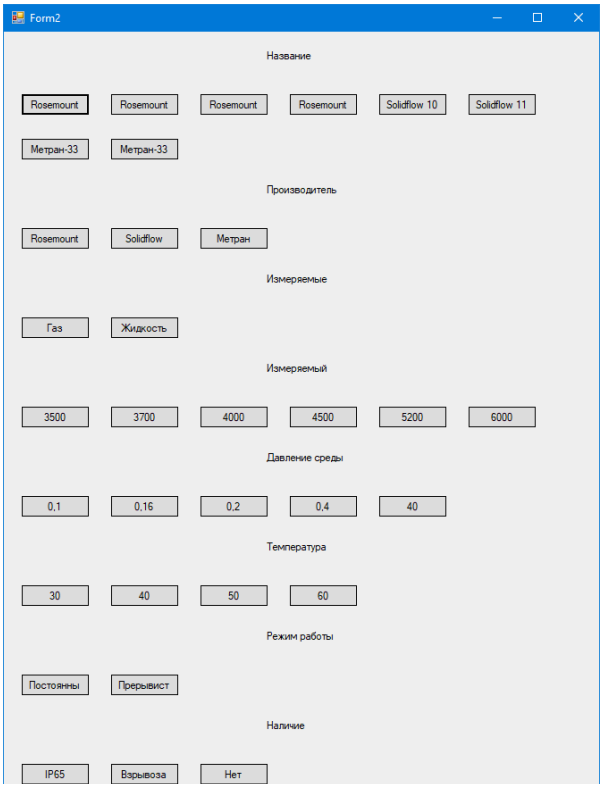


Рисунок12– Форма выбора СА

Form3

Данная форма предназначена для ввода пользовательского SQLзапроса на языке Trasnsact-SQL и изображена на рисунке 13. Форма содержит поле для ввода команд, а также кнопку для выполнения SQL запроса.

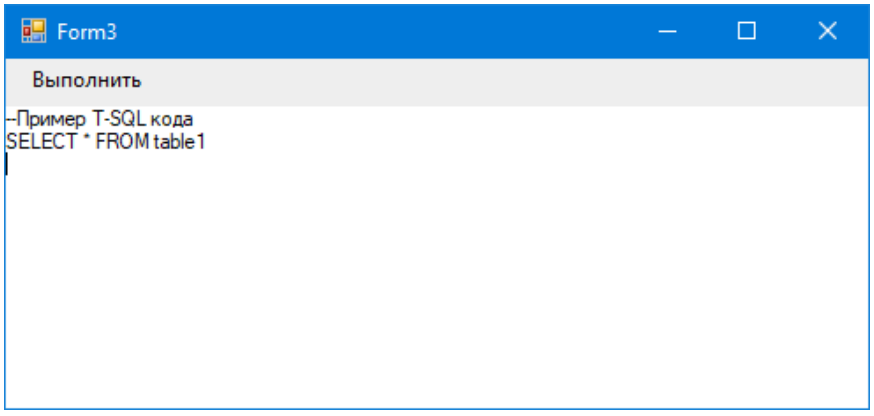


Рисунок13–Форма ввода пользовательского SQL запроса

Form4

Данная форма, изображенная на рисунке 14, предназначена для создания новой таблицы.Форма содержит:

– поле для ввода названия новой таблицы;

– поля таблицы для ввода названия, типа данных, размера, допустимого значения NULL, а также параметра по умолчанию.

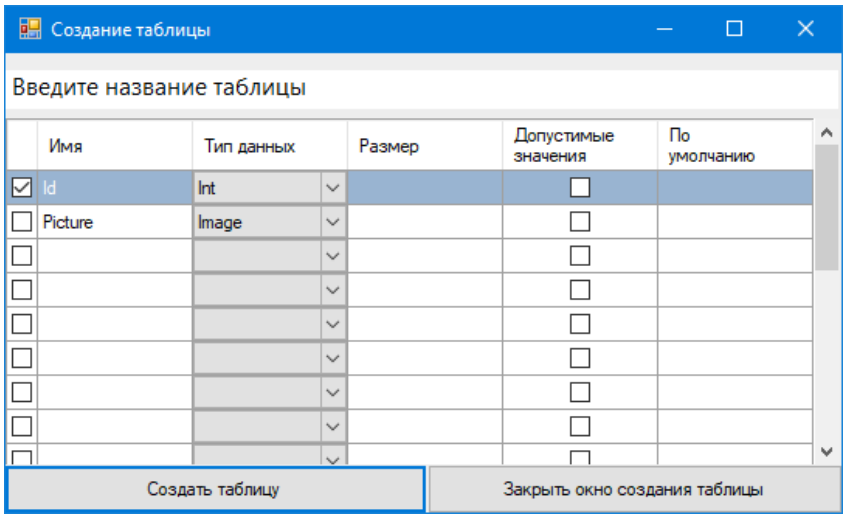


Рисунок 14 –Форма для создания новой таблицы

Form5

Форма, изображенная на рисунке 15, предназначена для отображения прикрепленных даташитов в виде изображения. Для того, чтобы вывести

изображение интересующего СА необходимо выбрать соответствующую строку из таблицы и удерживать на ней указатель мыши.

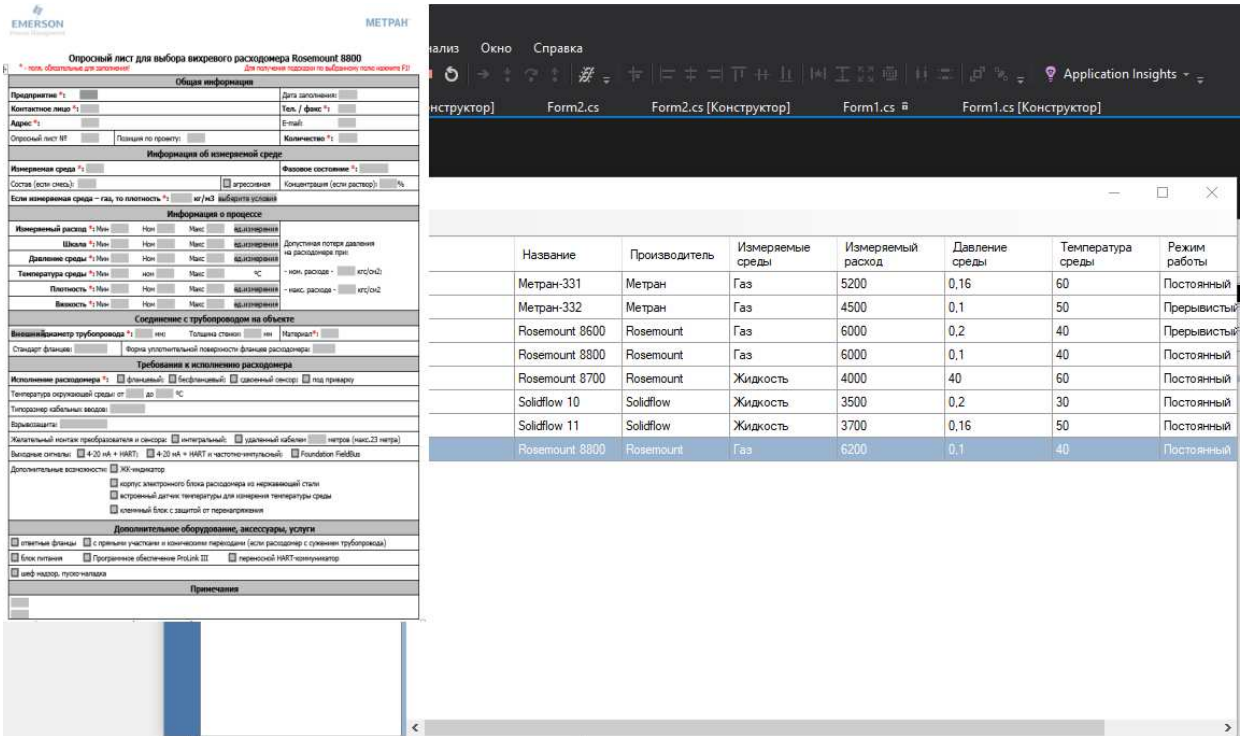


Рисунок15–Форма для отображения прикрепленный изображений

Form6

Форма, изображенная на рисунке 16, предназначена для добавления новой строки в определенную таблицу с СА. Данная форма содержит таблицу с полями:

– Id – номер строки в таблице;
– Picture – изображение даташита;
– Название – название модели СА;
– Производитель – название фирмы производителя СА;
– Измеряемые среды;
– Измеряемый расход;
– Давление среды;
– Температура среды;
– Режим работы СА;
– Реализация – приспособленность СА к определенным условиям эксплуатации.

Со временем эксплуатации программного решения структура таблиц может претерпеть изменения.

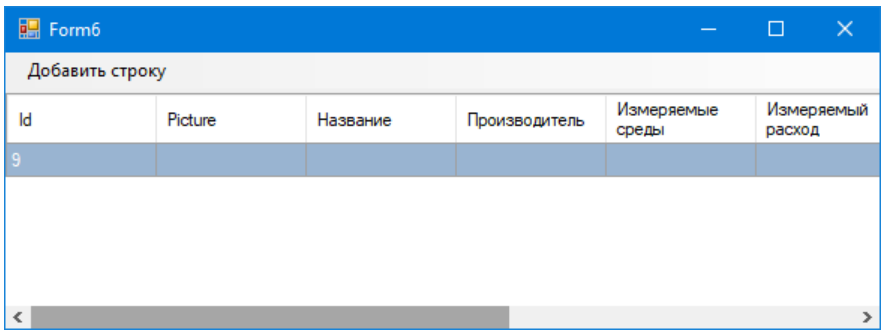


Рисунок 16 –Форма для добавления новой строки

Form7

Форма, изображенная на рисунке 17, предназначена для добавления нового столбца. Данная форма содержит таблицу с полями:

– Название столбца;

– Тип столбца – тип переменной, которая будет содержаться в столбце;

– Размер столбца – количество символов, которое будет выбранная ранее переменная.

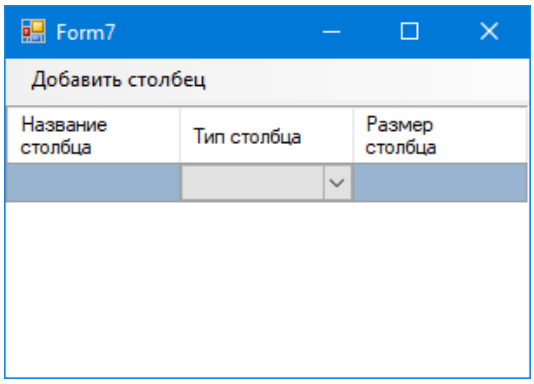


Рисунок 17 –Форма для добавления нового столбца

На рисунке 18 изображена таблица класса расходомеры, где находятся оставшиеся после выборки СА.

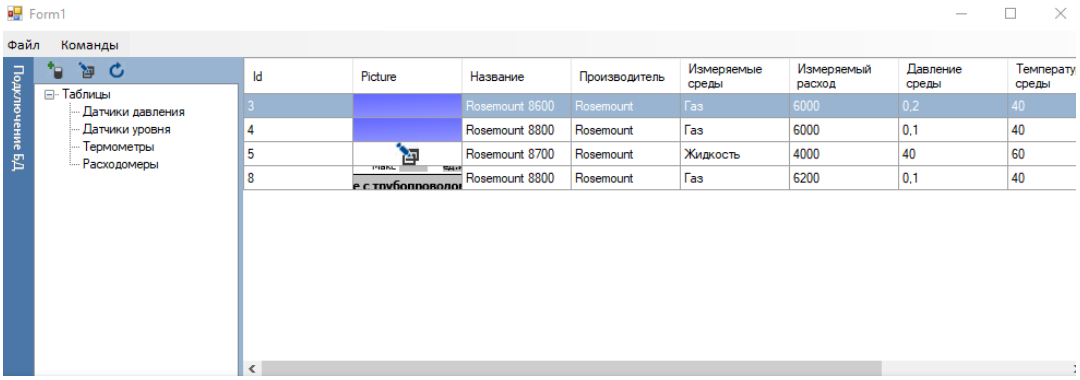
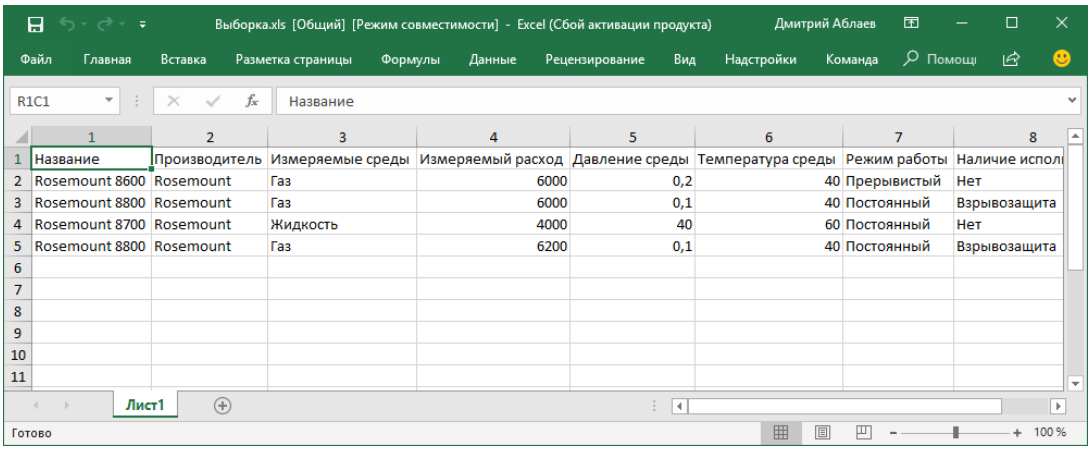


Рисунок 18 – Результат выборки СА

На рисунке 19 изображена выборка необходимых средств автоматизации, экспортированная в табличный документ MicrosoftExcel.



	1	2	3	4	5	6	7	8
1	Название	Производитель	Измеряемые среды	Измеряемый расход	Давление среды	Температура среды	Режим работы	Наличие испол
2	Rosemount 8600	Rosemount	Газ	6000	0,2	40	Прерывистый	Нет
3	Rosemount 8800	Rosemount	Газ	6000	0,1	40	Постоянный	Взрывозащита
4	Rosemount 8700	Rosemount	Жидкость	4000	40	60	Постоянный	Нет
5	Rosemount 8800	Rosemount	Газ	6200	0,1	40	Постоянный	Взрывозащита
6								
7								
8								
9								
10								
11								

Рисунок 19–Экспортированный в Excel файл результат выборки

Таким образом результатом работы созданного программного средства для выбора аппаратуры полевого уровня при проектировании АСУ ТП является перечень СА соответствующих условиям ТЗ, который в дальнейшем будет использован для решения задачи структурного синтеза АСУ ТП. Предложенное решение позволяет на начальном этапе формализовать проектную процедуру синтеза.

ЗАКЛЮЧЕНИЕ

В результате выпускной квалификационной была проанализирована предметная область, рассмотрены и детализированыобобщенные структуры, описывающие номенклатуру СА. На основе этого были проанализированы функциональные возможности будущего программного решения.

Разработано программное решение для поддержки выбора СА при проектированииАСУ ТП полевого уровняс учетом потребностей инженеров-проектировщиков. Реализована возможность оперативной проверки даташитов интересующего СА, а также экспорт информации в виде книги Excel.

Также апробированы возможности программного решения как СУБД, а именно:подключение внешней базы данных, создание новой таблицы, создание новых строк, столбцов, их удаление, вывод дополнительной визуальной информации о интересующем СА при наведении на соответствующую строку.

Реализованное программное средство для выбора аппаратуры полевого уровня при проектировании АСУ ТП формирует перечень СА соответствующих условиям ТЗ, который в дальнейшем будет использован для решения задачи структурного синтеза АСУ ТП. Предложенное решение позволяет на начальном этапе формализовать проектную процедуру синтеза.

СПИСОК СОКРАЩЕНИЙ

АСУ ТП – автоматизированная система управления технологическим процессом
СА – средства автоматизации
БД – база данных
САПР – система автоматизированного проектирования
Даташит – справочные листы с информацией

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Понятие проектирования [Электронный ресурс]. – Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=140_CADedu/CAD.cou/

2 Принципы системного подхода [Электронный ресурс]. – Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=190_CAD/0002.mod/?cou=140_CADedu/CAD.cou/

3 Стадии проектирования [Электронный ресурс]. – Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=190_CAD/0005.mod/?cou=140_CADedu/CAD.cou/

4 Уровни проектирования [Электронный ресурс]. – Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=190_CAD/0003.mod/?cou=140_CADedu/CAD.cou/

5 Общие сведения о САПР[Электронный ресурс]. – Режим доступа:http://bav-second.narod.ru/index2.html

6 Системы автоматизированного проектирования изделий и процессов: Учебное пособие/ Л.Р. Гирфанова.– Уфа: Уфимский государственный университет экономики и сервиса, 2014. – 143 с.

7 Основы теории и проектирования САПР : учеб. для втузов по спец. «Вычислительные машины, комплексы, системы и сети» / И. П. Норенков, В. Б. Маничев. – Москва : Высшая школа, 1990. – 335 с.

8 Программная реализация А - дерева для решения задач синтеза при проектировании АСУТП: Статья/Д.А. Аблаев, Д.А. Шапрун – Красноярск:СФУ, 2018. – 3 с.

9 Класс SQLConnection[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlconnection(v=vs.110).aspx

10 Класс SQLCommandBuilder[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqlcommandbuilder(v=vs.110).aspx

11 Класс SQLDataAdapter[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqldataadapter(v=vs.110).aspx

12 Класс DataTable [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.data.datatable(v=vs.110).aspx

13 Класс DataGridView [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.110).aspx

14 Класс MessageBox [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.messagebox(v=vs.110).aspx

15 Класс Form[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.form(v=vs.110).aspx

16 Класс SqlDataReader [Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.data.sqlclient.sqldatareader(v=vs.110).aspx
17 Класс TreeView[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.treeview(v=vs.110).aspx
18 Класс TreeNode[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.treenode(v=vs.110).aspx
19 Класс Point[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.drawing.point(v=vs.110).aspx
20 Класс OpenFileDialog[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.openfiledialog(v=vs.110).aspx
21 Класс Timer[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.timers.timer(v=vs.110).aspx
22 Класс MenuStrip[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.menustrip(v=vs.110).aspx
23 Класс ContextMenuStrip[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/system.windows.forms.contextmenustrip(v=vs.110).aspx
24 Интерфейс Application[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/microsoft.office.interop.excel.application.aspx
25 Объект Excelworkbook[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/vba/excel-vba/articles/workbook-object-excel
26 Объект Excelworksheet[Электронный ресурс]. – Режим доступа: https://msdn.microsoft.com/ru-ru/vba/excel-vba/articles/worksheet-object-excel

ПРИЛОЖЕНИЕ А

Исходный код Form1

```
usingSystem;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;

namespace WindowsFormsApp2
{
publicpartialclassForm1 : Form
   {
publicstring instr1;
static Image image;
      Form5 form5 = null;
      System.Drawing.Point mousePos;
staticpublicint matchR = 0;

private System.Drawing.Point mouseOffset;
privatebool isMouseDown = false;

publicstring[,] arrOfNames = null;
publicstring selStr = "SELECT * FROM table1 ";
publicstring strSql = "SELECT * FROM table1";
staticpublicstring defstrSql = "SELECT * FROM table1";
publicint lenOfStr;
publicint add1 = 0;
publicint cc = 0;
publicint rc1 = 0;
publicint foundedColumn = 0;
staticpublicstring str22 = System.IO.Path.GetFullPath("Database1.mdf");
publicstring conStr = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\dabla\Documents
\Visual Studio 2017\Projects\09022018\WindowsFormsApp2\Database1.mdf";
publicstring tableName = "table1"
```

```
static SqlConnection cn = new SqlConnection();
        SqlDataAdapter dataAdapter = null;
        System.Data.DataTable dataTable = null;
        SqlCommandBuilder bldr = null;

public SqlCommand myCommand = new SqlCommand("SELECT * FROM table1",
cn);
        SqlCommand select = newSqlCommand("SELECT * FROM table1", cn);

string cons;


public Form1()
        {

InitializeComponent();

        }

privatevoid Form1_Load(object sender, EventArgs e)
        {


lenOfStr = selStr.Length;

using (cn)
            {

                cn.ConnectionString = conStr;
try
                {
dataAdapter = new SqlDataAdapter();
bldr = new SqlCommandBuilder(dataAdapter);

                    dataAdapter.SelectCommand = myCommand;
                    dataAdapter.InsertCommand = bldr.GetInsertCommand();
                    dataAdapter.UpdateCommand = bldr.GetUpdateCommand();

cn.Open();

dataTable = new System.Data.DataTable();
dataAdapter.Fill(dataTable);
```

```
                dataGridView1.DataSource = dataTable;



            }
catch (SqlException ex)
            {
MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK);
            }
finally
            {
cn.Close();
            }

        }

    }

privatevoid GroupSelection(string string1, string strSql)
    {
        cn.ConnectionString = conStr;
using (cn)
        {
try
            {
dataGridView1.Columns.Clear();
cn.Open();

strSql = String.Format("SELECT [{1}].[{0}] FROM {1} GROUP BY [{1}].[{0}]",
string1, tableName);
myCommand = new SqlCommand(strSql, cn);



                SqlDataReader dataReader = myCommand.ExecuteReader();

                System.Data.DataTable dataTable = newSystem.Data.DataTable();

dataTable.Load(dataReader);

                dataGridView1.DataSource = dataTable;
dataReader.Close();
```

```
            }
catch (SqlException ex)
            {
MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK);
            }
finally
            {
cn.Close();
            }
        }
    }

publicvoid GroupSelection(string strSql)
    {
using (SqlConnection cn = new SqlConnection())
        {
            cn.ConnectionString = conStr;
            {
dataGridView1.Columns.Clear();
cn.Open();
myCommand = new SqlCommand(strSql, cn);


                SqlDataReader dataReader = myCommand.ExecuteReader();

                System.Data.DataTable dataTable = newSystem.Data.DataTable();

dataTable.Load(dataReader);

                dataGridView1.DataSource = dataTable;
dataReader.Close();

            }
cn.Close();
        }
    }

privatevoid button2_Click(object sender, EventArgs e)
    {
selStr = String.Format("SELECT * FROM {0} WHERE ", tableName);
обновитьToolStripMenuItem_Click(sender, e);
    }
```

```
privatevoid Btn1_Click(object sender, EventArgs e)
        {
for (int i = 0; i < cc; i++)
            {
if (((System.Windows.Forms.Button)sender).Name == arrOfNames[0, i])
                {
foundedColumn = i + 1;
                    button2.Text = foundedColumn.ToString();

                }
            }
        }

privatevoidвыходToolStripMenuItem_Click(object sender, EventArgs e)
        {
cn.Close();
System.Windows.Forms.Application.Exit();
        }

privatevoidиИлиДеревоToolStripMenuItem_Click(object sender, EventArgs e)
        {
defstrSql = String.Format("SELECT * FROM {0}", tableName);
GroupSelection(defstrSql);
int rc = dataGridView1.RowCount;
            cc = dataGridView1.Columns.Count - 2;


if (arrOfNames == null)
            {
arrOfNames = newstring[dataGridView1.RowCount + 1,
dataGridView1.Columns.Count - 2];
            }
else
            {
arrOfNames = null;
arrOfNames = newstring[dataGridView1.RowCount + 1,
dataGridView1.Columns.Count - 2];
            }

for (int i = 2; i < dataGridView1.Columns.Count; i++)
            {
arrOfNames[0, i - 2] = dataGridView1.Columns[i].Name;
```

```
        }
strSql = String.Format("SELECT * FROM {0}", tableName);
for (int i = 0; i < cc; i++)
        {
GroupSelection(arrOfNames[0, i], strSql);
for (int j = 0; j < dataGridView1.RowCount; j++)
            {
arrOfNames[j + 1, i] = dataGridView1.Rows[j].Cells[0].Value.ToString();
                rc1 = dataGridView1.RowCount;
            }

        }

strSql = String.Format("SELECT * FROM {0}", tableName);

GroupSelection(strSql);
        rc1 = dataGridView1.RowCount;
        Form2 form2 = newForm2(this);
form2.Show();

    }

privatevoidсоздатьЗапросToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form3 form3 = newForm3(this);
form3.Show();
    }

privatevoidсохранитьБазуДанныхToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        cn.ConnectionString = conStr;
using (cn)
        {
try
            {
cn.Open();

dataAdapter.Update(dataTable);
            }
catch (SqlException ex)
            {
```

```
MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK);
            }
finally
            {
cn.Close();
            }
        }
    }

privatevoidстрокуToolStripMenuItem_Click(object sender, EventArgs e)
    {
matchR = dataGridView1.RowCount + 1;
        Form6 form6 = newForm6(this);
form6.ShowDialog();




    }

privatevoidудалитьToolStripMenuItem_Click(object sender, EventArgs e)
    {
if (MessageBox.Show(String.Format("Выдействительнохотитеудалить {0}
строку?", dataGridView1.CurrentRow.Index+1), "Подтверждение",
MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            cn.ConnectionString = conStr;
using (cn)
            {
                SqlCommand delete = newSqlCommand(String.Format("DELETE
FROM {1} WHERE [id]={0}",
dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value,
tableName), cn);

                dataAdapter.DeleteCommand = delete;

cn.Open();
                dataAdapter.SelectCommand = myCommand;
dataAdapter.DeleteCommand.ExecuteNonQuery();

dataTable = new System.Data.DataTable();
dataAdapter.Fill(dataTable);
                dataGridView1.DataSource = dataTable;
```

```
cn.Close();
dbInr();
            }
        }
    }

void dbInr(bool a)
    {
for (int i = 0; i < dataGridView1.RowCount; i++)
        {
if (Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value) != i+1)
            {
if (MessageBox.Show(String.Format("Обнаруженнеправильныйиндекс: {0}.
Исправить?", i+1), "Ошибка!", MessageBoxButtons.YesNo) == DialogResult.Yes)
                {

for (int j = 1; j < dataGridView1.RowCount + 1; j++)
dataGridView1.Rows[j - 1].Cells[0].Value = j;
                }
            }
        }
    }

void dbInr()
    {

for (int j = 1; j < dataGridView1.RowCount + 1; j++)
dataGridView1.Rows[j - 1].Cells[0].Value = j;
    }

privatevoidобновитьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        cn.ConnectionString = conStr;
using (cn)
        {
cn.Open();
dataTable = new System.Data.DataTable();
dataAdapter.Fill(dataTable);
            dataGridView1.DataSource = dataTable;
cn.Close();
        }
    }
```

```
privatevoidподгрузитьБазуДанныхToolStripMenuItem_Click(object sender,
EventArgs e)
        {

if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {

cons = openFileDialog1.FileName;
conStr = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" + cons
+ "";

                SqlConnection conn = newSqlConnection();
                conn.ConnectionString = conStr;

using (conn)
                {
conn.Open();

string sql = "SELECT TABLE_NAME FROM information_schema.TABLES";

                    SqlCommand command = newSqlCommand(sql, conn);

                    SqlDataReader reader = command.ExecuteReader();

treeView1.BeginUpdate();
treeView1.Nodes.Clear();
treeView1.Nodes.Add("Таблицы");

while (reader.Read())
                    {
treeView1.Nodes[0].Nodes.Add(reader.GetString(0));
                    }

treeView1.EndUpdate();
treeView1.ExpandAll();

conn.Close();


                }
}
```

```
        }

privatevoid pictureBox1_MouseDown(object sender, MouseEventArgs e)
        {
int xOffset;
int yOffset;

if (e.Button == MouseButtons.Left)
            {
xOffset = -e.X - SystemInformation.FrameBorderSize.Width + 5;
yOffset = -e.Y - SystemInformation.FrameBorderSize.Height + 5;
mouseOffset = new System.Drawing.Point(xOffset, yOffset);
isMouseDown = true;
            }
        }

privatevoid customImageButton1_Click(object sender, EventArgs e)
        {
System.Windows.Forms.Application.Exit();
        }

privatevoid customImageButton2_Click(object sender, EventArgs e)
        {
if (this.WindowState == System.Windows.Forms.FormWindowState.Normal)
            {
this.WindowState = System.Windows.Forms.FormWindowState.Minimized;
            }
        }

privatevoid pictureBox1_MouseMove(object sender, MouseEventArgs e)
        {
if (isMouseDown)
            {
                System.Drawing.Point mousePos = Control.MousePosition;
mousePos.Offset(mouseOffset.X, mouseOffset.Y);
                Location = mousePos;
            }
 }

privatevoid pictureBox1_MouseUp(object sender, MouseEventArgs e)
        {
if (e.Button == MouseButtons.Left)
            {
```

```
isMouseDown = false;
            }
        }

publicstring SubStrDel(String str, string substr)
        {
int n = str.IndexOf(substr);
str = str.Remove(n, substr.Length);
return str;
        }

privatevoidподключениеБДToolStripMenuItem_Click(object sender, EventArgs e)
        {
if (splitContainer1.Panel1Collapsed == true)
                splitContainer1.Panel1Collapsed = false;
else
                splitContainer1.Panel1Collapsed = true;
        }

privatevoiddобToolStripMenuItem_Click(object sender, EventArgs e)
        {


if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {



cons = openFileDialog1.FileName;
conStr = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" + cons
+ "";

                SqlConnection conn = newSqlConnection();
                conn.ConnectionString = conStr;

using (conn)
                {
conn.Open();

string sql = "SELECT TABLE_NAME FROM information_schema.TABLES";

                    SqlCommand command = newSqlCommand(sql, conn);
```

```
                SqlDataReader reader = command.ExecuteReader();

treeView1.BeginUpdate();
treeView1.Nodes.Clear();
treeView1.Nodes.Add("Таблицы");

while (reader.Read())
                {
treeView1.Nodes[0].Nodes.Add(reader.GetString(0));
                }

treeView1.EndUpdate();
treeView1.ExpandAll();

conn.Close();


            }
        }
    }

privatevoid treeView1_AfterSelect(object sender, TreeViewEventArgs e)
    {

tableName = treeView1.Nodes[0].Nodes[treeView1.SelectedNode.Index].Text;
conStr = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" + cons
+ "";

using (cn)
            {

                cn.ConnectionString = conStr;
try
                {
dataAdapter = new SqlDataAdapter();
bldr = new SqlCommandBuilder(dataAdapter);
myCommand = new SqlCommand(String.Format("SELECT * FROM [{0}]",
tableName), cn);
                    dataAdapter.SelectCommand = myCommand;
                    dataAdapter.InsertCommand = bldr.GetInsertCommand();
                    dataAdapter.UpdateCommand = bldr.GetUpdateCommand();

cn.Open();
```

```
dataTable = new System.Data.DataTable();
dataAdapter.Fill(dataTable);
                dataGridView1.DataSource = dataTable;

            }
catch (SqlException ex)
            {
MessageBox.Show(ex.ToString(), "Error", MessageBoxButtons.OK);
            }
finally
            {
cn.Close();
            }

        }
dataGridView1.Columns[1].Visible = false;
dbInr(true);
    }

privatevoidдобавитьТаблицуToolStripMenuItem_Click(object sender, EventArgs e)
    {
if (treeView1.Nodes.Count != 0)
        {
            Form4 form4 = newForm4(this);
form4.ShowDialog();
refreshTableList();
        }
else
MessageBox.Show("Необходимоподключитьбазуданных!", "Ошибка!");

}

privatevoidобновитьToolStripMenuItem2_Click(object sender, EventArgs e)
    {
refreshTableList();
    }

privatevoidудалитьТаблицуToolStripMenuItem_Click(object sender, EventArgs e)
    {
if (treeView1.Nodes.Count != 0)
        {
```

```
string tableDelete = String.Format("DROP TABLE [dbo].[{0}]",
treeView1.Nodes[0].Nodes[treeView1.SelectedNode.Index].Text);

using (SqlConnection connection = new SqlConnection(conStr))
            {
connection.Open();
                SqlCommand command = newSqlCommand(tableDelete, connection);
                SqlDataReader reader = command.ExecuteReader();
reader.Close();
            }

refreshTableList();

        }
else
MessageBox.Show("Необходимоподключитьбазуданных!", "Ошибка!");
      }

void refreshTableList()
      {

            SqlConnection conn = newSqlConnection();
            conn.ConnectionString = conStr;

using (conn)
            {
conn.Open();

string sql = "SELECT TABLE_NAME FROM information_schema.TABLES";

                SqlCommand command = newSqlCommand(sql, conn);

                SqlDataReader reader = command.ExecuteReader();

treeView1.BeginUpdate();
treeView1.Nodes.Clear();
treeView1.Nodes.Add("Таблицы");

while (reader.Read())
                {
treeView1.Nodes[0].Nodes.Add(reader.GetString(0));
                }
```

```
treeView1.EndUpdate();
treeView1.ExpandAll();

conn.Close();


            }

        }

privatevoid dataGridView1_CellMouseEnter(object sender,
DataGridViewCellEventArgs e)
        {
            form5 = newForm5(this);


            mousePos.X = Control.MousePosition.X - form5.Size.Width;
            mousePos.Y = Control.MousePosition.Y - form5.Size.Height;

if (mousePos.X < 0 & mousePos.Y < 0)
            {

                mousePos.X = Control.MousePosition.X;
                mousePos.Y = Control.MousePosition.Y;
            }
else
            {
if (mousePos.Y < 0)
                {
                    mousePos.Y = 0;
                }
if (mousePos.X < 0)
                {
                    mousePos.X = 0;
                }
            }

            timer1.Enabled = true;

        }

privatevoid timer1_Tick(object sender, EventArgs e)
        {
```

```
            form5.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
form5.Show();
            form5.Location = mousePos;
        }

privatevoid dataGridView1_CellMouseLeave(object sender,
DataGridViewCellEventArgs e)
        {
            timer1.Enabled = false;
form5.Close();
        }

privatevoidстолбецToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form7 form7 = newForm7(this);
form7.ShowDialog();
        }

privatevoidудалитьСтолбецToolStripMenuItem_Click(object sender, EventArgs e)
        {
if (MessageBox.Show(String.Format("Выдействительнохотитеудалитьстолбец
{0}?", dataGridView1.Columns[dataGridView1.CurrentCell.ColumnIndex].Name),
"Подтверждение", MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                String delColumn = String.Format("ALTER TABLE [dbo].[{0}] DROP
COLUMN [{1}]", tableName,
dataGridView1.Columns[dataGridView1.CurrentCell.ColumnIndex].Name);

using (SqlConnection connection = new SqlConnection(conStr))
                {
connection.Open();
                    SqlCommand command = newSqlCommand(delColumn, connection);
                    SqlDataReader reader = command.ExecuteReader();
reader.Close();

                    System.Data.DataTable dataTable = newSystem.Data.DataTable();
                    SqlDataAdapter dataAdapter = newSqlDataAdapter();
                    SqlCommand cmd1 = newSqlCommand(String.Format("SELECT *
FROM {0}", tableName), connection);
                    dataAdapter.SelectCommand = cmd1;
dataAdapter.Fill(dataTable);
dataGridView1.DataSource = dataTable;
                }
```

```
        }
    }

privatevoidэкспортироватьТаблицуВExcelФайлToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        Microsoft.Office.Interop.Excel.Application Excel =
newMicrosoft.Office.Interop.Excel.Application();

        Workbook wb = Excel.Workbooks.Add(XlSheetType.xlWorksheet);

        Worksheet ws = (Worksheet)Excel.ActiveSheet;

for(int i = 2; i < dataGridView1.ColumnCount; i++)
        {
ws.Cells[1, i-1] = dataGridView1.Columns[i].Name;
for (int j = 1; j < dataGridView1.RowCount; j++)
            {
ws.Cells[j + 1, i-1] = dataGridView1.Rows[j].Cells[i].Value;
            }
        }
wb.SaveAs("Выборка.xls", XlFileFormat.xlWorkbookNormal,
   System.Reflection.Missing.Value, System.Reflection.Missing.Value, false, false,
   XlSaveAsAccessMode.xlShared, false, false, System.Reflection.Missing.Value,
   System.Reflection.Missing.Value, System.Reflection.Missing.Value);
Excel.Quit();

    }
  }
}
```

ИсходныйкодForm2

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

namespace WindowsFormsApp2
{
publicpartialclassForm2 : Form
   {

      Form1 frm1;

publicint ncounter;
public Point point1 = new Point(0, 25);
public Point point = new Point(20, 70);

public Form2(Form1 form1)
      {


InitializeComponent();
         frm1 = form1;

this.BackColor = frm1.BackColor;
         point1.X = frm1.Size.Width / 2;
         point1.Y = 20;
         point1.X = this.Size.Width / 2;
for (int j = 0; j < frm1.cc; j++)
         {

for (int i = 0; i < frm1.rc1+1; i++)
            {
if (frm1.arrOfNames[i, j] == null)
               {

continue;
               }
```

```
else
                {
if (i == 0)
                    {
CreateLabel(i, j);

                    }
else
                    {
CreateButton(i, j);

                    }
                }

            }
            point.X = 20;
            point1.Y = point.Y + 50;
            point.Y = point.Y + 100;

        }

    }

privatevoid Btn1_Click(object sender, EventArgs e)
    {


for (int i = 1; i < frm1.rc1+1; i++)
        {

for (int j = 0; j < frm1.cc; j++)
            {
if (frm1.arrOfNames[i, j] == null)
                {
continue;
                }
if (((Button)sender).Name == frm1.arrOfNames[i, j])
                {
                    String temp = frm1.arrOfNames[i, j];
if(temp.Contains(",") == true)
                        {
temp = temp.Replace(",", ".");
                        }
```

```
if(((Button)sender).BackColor.Equals(System.Drawing.Color.Gainsboro))
                    {
                        ((Button)sender).BackColor = System.Drawing.Color.DarkGray;
if (frm1.lenOfStr == frm1.selStr.Length)
                        {
                            frm1.selStr = String.Format("SELECT * FROM {0}
",frm1.tableName) + String.Format("WHERE {2}.[{0}] = N'{1}'",
frm1.arrOfNames[0, j], temp,frm1.tableName);
                        }
else
                        {
                            frm1.selStr = frm1.selStr + String.Format(" and {2}.[{0}] =
N'{1}'", frm1.arrOfNames[0, j], temp, frm1.tableName);
ncounter++;
                        }
                    }
else
                    {
                        ((Button)sender).BackColor = System.Drawing.Color.Gainsboro;
if(ncounter>0)
                        {
                            frm1.selStr = frm1.SubStrDel(frm1.selStr, String.Format(" and
{2}.[{0}] = N'{1}'", frm1.arrOfNames[0, j], temp, frm1.tableName));
ncounter--;
                        }
else
                        {
                            frm1.selStr = frm1.SubStrDel(frm1.selStr,
String.Format("{2}.[{0}] = N'{1}'", frm1.arrOfNames[0, j], temp, frm1.tableName));
                            frm1.selStr = frm1.SubStrDel(frm1.selStr,"WHERE ");
                        }
                    }
                    frm1.button2.Text = frm1.arrOfNames[i,j].ToString();
frm1.GroupSelection(frm1.selStr);
                }
            }
        }
    }

privatevoid Form2_Load(object sender, EventArgs e)
    {
if (frm1.WindowState != FormWindowState.Maximized)
```

```
        {
            Point startPosition = newPoint(frm1.Location.X + frm1.Size.Width - 15,
frm1.Location.Y);
this.Location = startPosition;
        }
ncounter = 0;
        frm1.selStr = String.Format("SELECT * FROM {0} ", frm1.tableName);
        frm1.lenOfStr = frm1.selStr.Length;

    }

void CreateLabel(int i, int j)
    {
        Label l1 = newLabel();



        l1.Text = frm1.arrOfNames[i, j];
        point1.X = this.Size.Width / 2 - l1.Size.Width / 2;
        l1.Location = point1;

Controls.Add(l1);
    }

void CreateButton(int i, int j)
    {
        frm1.add1 = 100;
if (point.X >this.Size.Width - 100)
        {
            point.Y = point.Y + 50;
            point.X = 20;
        }

        Button btn1 = newButton();
        btn1.Text = frm1.arrOfNames[i, j].ToString();
        btn1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        btn1.BackColor = System.Drawing.Color.Gainsboro;
        btn1.TextAlign = ContentAlignment.MiddleCenter;

        btn1.Name = frm1.arrOfNames[i, j].ToString();
        btn1.Location = point;
this.Controls.Add(btn1);
        btn1.Click += Btn1_Click;
```

```
            point.X = point.X + frm1.add1;
        }


    }

}
```

ИсходныйкодForm3

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

namespace WindowsFormsApp2
{
publicpartialclassForm3 : Form
   {
      Form1 frm1;
public Form3(Form1 form1)
      {
         frm1 = form1;
InitializeComponent();
      }

privatevoid button1_Click(object sender, EventArgs e)
      {

      }

privatevoidвыполнитьToolStripMenuItem_Click(object sender, EventArgs e)
      {
if (textBox1.Text != "")
         {
string strSql = textBox1.Text;
frm1.GroupSelection(strSql);
         }
else
         {
MessageBox.Show("Необходимоввестикоманду SQL!", "Ошибка!",
MessageBoxButtons.OK);
         }
      }

privatevoid Form3_Load(object sender, EventArgs e)
      {
```

```
this.BackColor = frm1.BackColor;
this.menuStrip1.BackColor = frm1.menuStrip1.BackColor;
this.menuStrip1.RenderMode = frm1.menuStrip1.RenderMode;
this.menuStrip1.Font.Name.Replace(this.menuStrip1.Font.Name,
frm1.menuStrip1.Font.Name);

      }
   }
}
```

ИсходныйкодForm4

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

namespace WindowsFormsApp2
{
publicpartialclassForm4 : Form
   {
      Form1 frm1;

public Form4(Form1 form1)
      {
InitializeComponent();
         frm1 = form1;

      }

privatevoid button3_Click(object sender, EventArgs e)
      {

      }

privatevoid button1_Click(object sender, EventArgs e)
{
if (textBox1.Text == "Введите название таблицы")
         {
            MessageBox.Show("Необходимо ввести название таблицы",
"Ошибка!");
}
else
         {
addTable(frm1.conStr, textBox1.Text);
this.Close();
         }
      }
```

```
privatevoid button2_Click(object sender, EventArgs e)
        {
this.Close();
        }

privatevoid Form4_Load(object sender, EventArgs e)
        {
           DataGridViewComboBoxCell Cb = newDataGridViewComboBoxCell();
Cb.Items.Add("BigInt");
Cb.Items.Add("Binary");
Cb.Items.Add("Bit");
Cb.Items.Add("Char");
Cb.Items.Add("Date");
Cb.Items.Add("DateTime");
Cb.Items.Add("DateTime2");
Cb.Items.Add("DateTimeOffset");
Cb.Items.Add("Decimal");
Cb.Items.Add("Float");
Cb.Items.Add("Image");
Cb.Items.Add("Int");
Cb.Items.Add("Money");
Cb.Items.Add("NChar");
Cb.Items.Add("NText");
Cb.Items.Add("NVarChar");
Cb.Items.Add("Real");
Cb.Items.Add("SmallDateTime");
Cb.Items.Add("SmallInt");
Cb.Items.Add("SmallMoney");
Cb.Items.Add("Structured");
Cb.Items.Add("Text");
Cb.Items.Add("Time");
Cb.Items.Add("Timestamp");
Cb.Items.Add("TinyInt");
Cb.Items.Add("Udt");
Cb.Items.Add("UniqueIdentifier");
Cb.Items.Add("VarBinary");
Cb.Items.Add("VarChar");
Cb.Items.Add("Variant");
Cb.Items.Add("Xml");


dataGridView1.Columns[2].CellTemplate = Cb;
```

```
dataGridView1.Rows.Add(20);
dataGridView1.Rows[0].Cells[0].Value = true;
dataGridView1.Rows[0].Cells[1].Value = "Id";
dataGridView1.Rows[0].Cells[2].Value = "Int";
dataGridView1.Rows[0].Cells[4].Value = null;

dataGridView1.Rows[1].Cells[1].Value = "Picture";
dataGridView1.Rows[1].Cells[2].Value = "Image";
      }

void addTable(string conStr, string tableName)
      {

string tableCreate = String.Format("CREATE TABLE [dbo].[{0}](",tableName);
for (int i = 0; i < dataGridView1.RowCount; i++)
          {
if (dataGridView1.Rows[i].Cells[1].Value == null)
continue;
if (i > 0)
tableCreate += ",";
if (dataGridView1.Rows[i].Cells[0].Value != null)
              {

tableCreate += String.Format("PRIMARY KEY ([{0}]), ",
dataGridView1.Rows[i].Cells[1].Value);

              }
if (dataGridView1.Rows[i].Cells[3].Value != null)
              {
tableCreate += String.Format("[{0}] {1}({2}) ",
dataGridView1.Rows[i].Cells[1].Value, dataGridView1.Rows[i].Cells[2].Value,
dataGridView1.Rows[i].Cells[3].Value);
}
else
              {
tableCreate += String.Format("[{0}] {1} ", dataGridView1.Rows[i].Cells[1].Value,
dataGridView1.Rows[i].Cells[2].Value);
              }
if (dataGridView1.Rows[i].Cells[4].Value != null)
              {
tableCreate += "NULL ";
              }
```

```
else
            {
tableCreate += "NOT NULL ";
            }
tableCreate += String.Format("{0}", dataGridView1.Rows[i].Cells[5].Value);

        }
        tableCreate = tableCreate + ");";

using (SqlConnection connection = new SqlConnection(conStr))
        {
connection.Open();
            SqlCommand command = newSqlCommand(@tableCreate, connection);
            SqlDataReader reader = command.ExecuteReader();
reader.Close();
        }


    }

privatevoid textBox1_MouseClick(object sender, MouseEventArgs e)
    {
        textBox1.Text = "";
    }
  }


}
```

ИсходныйкодForm5

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

namespace WindowsFormsApp2
{
publicpartialclassForm5 : Form
   {
      Form1 frm1;
public Form5(Form1 form1)
      {
InitializeComponent();
         frm1 = form1;
      }

privatevoid Form5_Load(object sender, EventArgs e)
      {

         ImageConverter ic = newImageConverter();

         Image img =
(Image)ic.ConvertFrom(frm1.dataGridView1.Rows[frm1.dataGridView1.CurrentRo
w.Index].Cells[1].Value);

         Bitmap bitmap1 = newBitmap(img);


         pictureBox1.Image = bitmap1;
      }
   }
}
```

ИсходныйкодForm6

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

namespace WindowsFormsApp2
{
publicpartialclassForm6 : Form
   {
      Form1 frm1;
string filename = "";
byte[] imageData;
bool match = false;
int[] arrofint = newint[999];

public Form6(Form1 form1)
      {
InitializeComponent();
         frm1 = form1;
      }

privatevoid Form6_Load(object sender, EventArgs e)
      {
using (SqlConnection connection = new SqlConnection(frm1.conStr))
         {


               DataTable dataTable = newDataTable();
               SqlDataAdapter dataAdapter = newSqlDataAdapter();

               SqlCommand myCommand =
newSqlCommand(String.Format("SELECT * FROM {0}", frm1.tableName),
connection);
               dataAdapter.SelectCommand = myCommand;
connection.Open();
```

```
dataAdapter.Fill(dataTable);
                frm1.dataGridView1.DataSource = dataTable;
connection.Close();

        }

filename = "";

for (int i = 0; i < frm1.dataGridView1.ColumnCount; i++)
        {
dataGridView1.Columns.Add(String.Format("Column {0}", i),
frm1.dataGridView1.Columns[i].HeaderText);
        }


for(int i = 0; i< frm1.dataGridView1.RowCount; i++)
        {
if ((int)frm1.dataGridView1.Rows[i].Cells[0].Value ==
frm1.dataGridView1.RowCount)
            {
match = true;
break;
            }
else
            {
match = false;
            }
        }

if (match == false)
        {
dataGridView1.Rows[0].Cells[0].Value = frm1.dataGridView1.RowCount + 1;
        }
else
        {
for(int i = 0; i < frm1.dataGridView1.RowCount; i++)
            {
arrofint[i] = Convert.ToInt32(frm1.dataGridView1.Rows[i].Cells[0].Value);
            }
dataGridView1.Rows[0].Cells[0].Value = arrofint.Max()+1;

        }
```

```
    }

privatevoidсохранитьToolStripMenuItem_Click(object sender, EventArgs e)
{
if (filename == "")
        {
          MessageBox.Show("Перед добавлением новой строки необходимо
добавить изображение", "Внимание!");

if (getImageData() == System.Windows.Forms.DialogResult.OK)
          {
            frm1.instr1 = String.Format("INSERT INTO {0} VALUES(",
frm1.tableName);

for (int i = 0; i < frm1.dataGridView1.ColumnCount; i++)
            {

if (i != 1)
              {
                frm1.instr1 += String.Format("'{0}'",
dataGridView1.Rows[0].Cells[i].Value);
              }
else
              {
                frm1.instr1 += "@picture";
              }
if (i != frm1.dataGridView1.ColumnCount - 1)
              {
                frm1.instr1 += ", ";
              }

            }
            frm1.instr1 += ")";


using (SqlConnection connection = new SqlConnection(frm1.conStr))
            {
connection.Open();
              SqlCommand command = newSqlCommand();
              command.Connection = connection;
```

```
                command.CommandText = @frm1.instr1;

command.Parameters.Add("@picture", SqlDbType.Image, 1000000);

                command.Parameters["@picture"].Value = imageData;

command.ExecuteNonQuery();
              }
this.Close();
            }
          }
else
          {
            frm1.instr1 = String.Format("INSERT INTO {0} VALUES(",
frm1.tableName);

for (int i = 0; i < frm1.dataGridView1.ColumnCount; i++)
            {

if (i != 1)
              {
                frm1.instr1 += String.Format("N'{0}'",
dataGridView1.Rows[0].Cells[i].Value);
              }
else
              {
                frm1.instr1 += "@picture";
              }
if (i != frm1.dataGridView1.ColumnCount - 1)
              {
                frm1.instr1 += ", ";
              }

            }
            frm1.instr1 += ")";

using (SqlConnection connection = new SqlConnection(frm1.conStr))
            {
connection.Open();
              SqlCommand command = newSqlCommand();
              command.Connection = connection;

              command.CommandText = frm1.instr1;
```

```
command.Parameters.Add("@picture", SqlDbType.Image, 1000000);

                command.Parameters["@picture"].Value = imageData;

command.ExecuteNonQuery();

                DataTable dataTable = newDataTable();
                SqlDataAdapter dataAdapter = newSqlDataAdapter();
                SqlCommand cmd1 = newSqlCommand(String.Format("SELECT *
FROM {0}", frm1.tableName), connection);
                dataAdapter.SelectCommand = cmd1;
dataAdapter.Fill(dataTable);
                frm1.dataGridView1.DataSource = dataTable;
            }
        }
    }


privatevoid dataGridView1_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
    {
getImageData();

    }

    System.Windows.Forms.DialogResult getImageData()
    {

        System.Windows.Forms.DialogResult dialogResult =
openFileDialog1.ShowDialog();
if (dialogResult == System.Windows.Forms.DialogResult.OK)
        {

filename = openFileDialog1.FileName;

using (System.IO.FileStream fs = new System.IO.FileStream(filename,
FileMode.Open))
            {
imageData = newbyte[fs.Length];
fs.Read(imageData, 0, imageData.Length);
            }
```

```
            }
return dialogResult;
        }
    }
}
```

ИсходныйкодForm7

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

namespace WindowsFormsApp2
{
publicpartialclassForm7 : Form
   {
      Form1 frm1;

      String addColumn = "";
public Form7(Form1 form1)
      {
InitializeComponent();
         frm1 = form1;
      }

privatevoid Form7_Load(object sender, EventArgs e)
      {
         DataGridViewComboBoxCell Cb = newDataGridViewComboBoxCell();
Cb.Items.Add("BigInt");
Cb.Items.Add("Binary");
Cb.Items.Add("Bit");
Cb.Items.Add("Char");
Cb.Items.Add("Date");
Cb.Items.Add("DateTime");
Cb.Items.Add("DateTime2");
Cb.Items.Add("DateTimeOffset");
Cb.Items.Add("Decimal");
Cb.Items.Add("Float");
Cb.Items.Add("Image");
Cb.Items.Add("Int");
Cb.Items.Add("Money");
Cb.Items.Add("NChar");
```

```
Cb.Items.Add("NText");
Cb.Items.Add("NVarChar");
Cb.Items.Add("Real");
Cb.Items.Add("SmallDateTime");
Cb.Items.Add("SmallInt");
Cb.Items.Add("SmallMoney");
Cb.Items.Add("Structured");
Cb.Items.Add("Text");
Cb.Items.Add("Time");
Cb.Items.Add("Timestamp");
Cb.Items.Add("TinyInt");
Cb.Items.Add("Udt");
Cb.Items.Add("UniqueIdentifier");
Cb.Items.Add("VarBinary");
Cb.Items.Add("VarChar");
Cb.Items.Add("Variant");
Cb.Items.Add("Xml");

dataGridView1.Columns[1].CellTemplate = Cb;
dataGridView1.Rows.Add();
      }

privatevoidсохранитьToolStripMenuItem_Click(object sender, EventArgs e)
      {
if (dataGridView1.Rows[0].Cells[2].Value == null)
          {
addColumn = String.Format("ALTER TABLE [dbo].[{0}] ADD {1} {2} NULL",
frm1.tableName, dataGridView1.Rows[0].Cells[0].Value,
dataGridView1.Rows[0].Cells[1].Value);
          }
else
          {
             addColumn = String.Format("ALTER TABLE [dbo].[{0}] ADD {1}
{2}({3}) NULL", frm1.tableName, dataGridView1.Rows[0].Cells[0].Value,
dataGridView1.Rows[0].Cells[1].Value, dataGridView1.Rows[0].Cells[2].Value);
          }



using (SqlConnection connection = new SqlConnection(frm1.conStr))
          {
connection.Open();
```

```
            SqlCommand command = newSqlCommand(addColumn, connection);
            SqlDataReader reader = command.ExecuteReader();
reader.Close();

            DataTable dataTable = newDataTable();
            SqlDataAdapter dataAdapter = newSqlDataAdapter();
            SqlCommand cmd1 = newSqlCommand(String.Format("SELECT * FROM
{0}", frm1.tableName), connection);
            dataAdapter.SelectCommand = cmd1;
dataAdapter.Fill(dataTable);
            frm1.dataGridView1.DataSource = dataTable;
        }


    }

  }
}
```

Федеральное государственное
автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра «Системы автоматики, автоматизированное
управление и проектирование»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.В. Ченцов
«16» 06 2018 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.04 – Управление в технических системах

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ВЫБОРА АППАРАТУРЫ ПОЛЕВОГО УРОВНЯ ПРИ ПРОЕКТИРОВАНИИ АСУ ТП

Руководитель _____ «15» 06 2018 г. доц., канд. техн. наук Е.Е. Носкова

Выпускник _____ «15» 06 2018 г. Д.А. Аблаев

Нормоконтролер _____ «15» 06 2018 г. Т.А. Грудинова

Красноярск 2018