ФГБОУ ВПО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

**Кафедра системного анализа и обработки информации**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине: Системный анализ

на тему: Разработка подсистемы для оценки организационных структур организации, основанной на информационном подходе

выполнил студент группы БИ1301

Грибков Михаил Евгеньевич

Допущен к защите

Руководитель проекта к.э.н., доцент Вострокнутов Александр Евгеньевич

Нормоконтролер к.э.н., доцент Вострокнутов Александр Евгеньевич

Защищен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии: д.э.н., профессор Барановская Т.П.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Краснодар-2016

ФГБОУ ВПО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

Кафедра системного анализа и обработки информации

УТВЕРЖДАЮ:

Зав. Кафедрой системного анализа и обработки информации

д.э.н., профессор Барановская Т.П.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗАДАНИЕ

на курсовой проект

Студенту группы БИ-1301 курса 3

Факультета прикладная информатика

Направления подготовки 38.03.05 Бизнес-информатика

Грибкову Михаилу Евгеньевичу

Тема работы: Разработка подсистемы для оценки организационных структур организации основанной на информационном подходе

Содержание задания: Введение; 1 Характеристики объекта исследования; 2 Разработка требований к программного обеспечения по оценке организационных структур; 3 Разработка программного обеспечения по оценке организационных структур; Список использованных источников

Объем работы:

а) пояснительная записка к проекту\_\_\_\_\_\_63\_\_\_\_\_\_\_\_\_листов формата А4

б) графическая часть\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-\_\_\_\_\_\_\_\_\_\_\_\_\_лист формата А4

Рекомендуемая литература: Гриценко Ю.Б. Архитектура предприятия [Электронный ресурс]: учебное пособие/ Гриценко Ю.Б.— Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2011. — 264 c.— Режим доступа: http://www.iprbookshop.ru/14005.— ЭБС «IPRbooks»

Срок выполнения проекта: с 1.03.2016г. по 17.06.2016г.

Срок защиты: 17.06.2016г.

Дата выдачи задания: 1.03.2016г.

Дата сдачи проекта на кафедру: 09.06.2016г.

Руководитель проекта к.э.н., доцент Вострокнутов Александр Евгеньевич.

(подпись, Ф.И.О., звание, степень)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

**Краснодар – 2016**

# ВВЕДЕНИЕ

С каждым днем появляются все новые предприятия, организации, становится жестче законодательство, на фоне чего конкурентная борьба приобретает все более жесткие формы. Затрудняется разработка и реализация стратегических планов организаций, которые, в свою очередь, требуют повышения ее гибкости, способности оперативно реагировать на постоянные меняющиеся условия деятельности, что обусловленно построением и функционированием структуры управления предприятием. Поэтому одним из основополагающих принципов рыночной экономики является отбор наиболее эффективных организационных систем.

Актуальность курсового проекта обусловлена тем, что оптимизация организационных структур управления является составной частью комплекса мер по повышению эффективности работы любого предприятия. Руководители должны выбрать ту структуру, которая лучше всего отвечает поставленным целям и задачам предприятия, своевременно и адекватно реагирует на действия факторов внутренней и внешней среды, целенаправленно распределяет и координирует усилия сотрудников и таким образом повышает собственную конкурентоспособность.

Целью данного проекта является разработка подистемы для оценки организационных структур организации, основанной на информационном подходе.

Задачи на курсовой проект:

* исследование предметной области;
* анализ объекта автоматизации;
* обоснованность выбора средств разработки;
* описание структуры базы данных;
* реализация бизнес-приложения;
* тестирование бизнес-приложения;
* составление руководство пользователя.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc453704494)

[1 ХАРАКТЕРИСТИКА ОБЪЕКТА ИССЛЕДОВАНИЯ И ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc453704495)

[1.1 Понятие организационных структур 5](#_Toc453704496)

[1.2 Сущность и назначение оценки организационных структур 8](#_Toc453704497)

[1.3 Методики оценки организационных структур 13](#_Toc453704498)

[2 РАЗРАБОТКА ТРЕБОВАНИЙ К ПО ПРОЦЕДУРЫ ОЦЕНКИ ОРГСТРУКТУР 21](#_Toc453704499)

[2.1 Разработка функциональной модели 21](#_Toc453704500)

[2.2 Выбор среды реализации ПО 22](#_Toc453704501)

[3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО ОЦЕНКЕ ОРГСТРУКТУР 24](#_Toc453704502)

[3.1 Описание процесса разработки 24](#_Toc453704503)

[3.2 Инструкция пользователя 26](#_Toc453704504)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc453704505)

[ПРИЛОЖЕНИЕ 33](#_Toc453704506)

# 1 ХАРАКТЕРИСТИКА ОБЪЕКТА ИССЛЕДОВАНИЯ И ПРЕДМЕТНОЙ ОБЛАСТИ

## Понятие организационных структур

**Организационная структура управления** — это упорядоченная совокупность взаимосвязанных элементов, находящихся между собой в устойчивых отношениях, обеспечивающих их развитие и функционирование как единого целого.

Организационная структура управления направлена на установление четких взаимосвязей между отдельными подразделениями организации, распределение между ними прав и ответственности.

Связи между элементами структуры управления бывают:

1) вертикальными, когда происходит взаимодействие между руководителем и подчиненным (например, связь между директором фирмы и управляющим структурным подразделением);

2) горизонтальными, когда происходит взаимодействие равноправных элементов (например, связи между управляющими структурными подразделениями одного уровня)

Виды отношений внутри организации аналогичны типу построения структуры ее управления и делятся на:

1) линейные отношения — это отношения между руководителем и его подчиненными;

2) функциональные отношения — это отношения специалиста, который уполномочен выполнять ту или иную функцию в рамках всей организации, с другими членами организации;

3) отношения управленческого аппарата, данный тип отношений имеет место в случае представления чьих-либо прав и полномочий. Должностные обязанности при этом состоят в предоставлении рекомендации, советов.

Структура управления оказывает огромное влияние на все стороны управления, так как связана с ключевыми понятиями менеджмента — целями, функциями, процессом, механизмом, функционирования, полномочиями людей. Поэтому менеджеры всех уровней уделяют огромное внимание принципам и методам формирования структур, выбору типа или комбинаций типов структур, изучению тенденций их построения, оценке их соответствия решаемым целям и задачам.

Элементами структуры управления являются: работник управления — человек, выполняющий определенную функцию управления; орган управления — группа работников, связанных определенными отношениями, состоящая из первичных групп.

Первичная группа — коллектив работников управления, у которого есть общий руководитель, но нет подчиненных.

Структура управления должна отражать цели и задачи фирмы, быть подчиненной производству и меняться вместе с ним. Она должна отражать функциональное разделение труда и объем полномочий работников управления; последние определяются политикой, процедурами, правилами и должностными инструкциями и расширяются, как правило, в направлении более высоких уровней управления. Полномочия руководителей ограничиваются факторами внешней среды, уровнем культуры и ценностными ориентациями, принятыми традициями и нормами. Структура управления должна удовлетворять множеству требований, которые отражают ее значение для менеджмента. Эти принципы учитываются в принципах проектирования организационной структуры управления. Принципы проектирования организационной структуры:

1. структура должна отражать цели и задачи организации, быть подчиненной производству и его потребностям;

2. структура должна предусматривать оптимальное разделение труда между органами управления и отдельными работниками, обеспечивающее творческий характер работы и нормальную нагрузку, а также надлежащую специализацию;

3. формирование структуры должно быть неразрывно с определением полномочий и ответственности каждого работника и органа управления, с установлением системы вертикальных и горизонтальных связей между ними;

4. структура должна поддерживать соответствие между функциями, обязанностями, полномочиями и ответственностью, так как нарушение его приводит к дисбалансу системы управления в целом;

5. структура управления должна быть адекватной социально-культурной среде организации, оказывать существенное влияние на решения относительно уровня централизации и детализации, распределения полномочий и ответственности, степени самостоятельности и масштабов контроля руководителей и менеджеров.

Важнейшими требованиями, которым должны удовлетворять организационные структуры управления являются следующие.

1. Оптимальность. Структура управления признается оптимальной, если между звеньями и ступенями управления на всех уровнях устанавливаются рациональные связи при наименьшем числе ступеней управления.

2. Оперативность. Суть данного требования состоит в том, чтобы за время от принятия решения до его исполнения в управляемой системе не успели произойти необратимые отрицательные изменения, делающие ненужной реализацию принятых решений.

3. Надежность. Структура аппарата управления должна гарантировать достоверность передачи информации, не допускать искажений управляющих команд и других передаваемых данных, обеспечивать бесперебойность связи в системе управления.

4. Экономичность. Задача состоит в том, чтобы нужный эффект от управления достигался при минимальных затратах на управленческий аппарат. Критерием этого может служить соотношение между затратами ресурсов и полезным результатом.

5. Гибкость. Способность изменяться в соответствии с изменениями внешней среды.

6. Устойчивость структуры управления. Неизменность ее основных свойств при различных внешних воздействиях, целостность функционирования системы управления и ее элементов.

## Сущность и назначение оценки организационных структур

Под структурой управления понимается упорядоченная совокупность устойчиво взаимосвязанных элементов, обеспечивающих функционирование и развитие организации как единого целого. Организационная структура управления (ОСУ) определяется также как форма разделения и кооперации управленческой деятельности, в рамках которой осуществляется процесс управления по соответствующим функциям, направленным на решение поставленных задач и достижение намеченных целей. С этих позиций организационная структура представляется в виде системы оптимального распределения функциональных обязанностей, прав и ответственности, порядка и форм взаимодействия между входящими в ее состав органами управления и работающими в них людьми.

Для каждой организации существует наилучшая и только ей присущая организационная структура производства и управления. Любая организация имеет некоторые особенности использования технологического оборудования, профессионализма и личностных качеств персонала, порядков и традиций между работниками по вертикали и горизонтали. Каждая организация имеет свою историю, культуру, технологическую философию и персонал. Для осуществления их гармонического взаимодействия и служит единственная, наиболее подходящая к ним организационная структура управления. Оригинальность конкретной организационной структуры управления достигается на базе использования существующих типов линейных, функциональных, штабных, и других структур путем включения в них или исключения из них каких-либо подразделений или связей.

Ключевыми понятиями структур управления являются элементы, связи (отношения), уровни и полномочия. Элементами ОСУ могут быть как отдельные работники (руководители, специалисты, служащие), так и службы, либо органы аппарата управления, в которых занято то или иное количество специалистов, выполняющих определенные функциональные обязанности. Существуют два направления специализации элементов ОСУ:

а) в зависимости от состава структурных подразделений организации вычленяются звенья структуры управления, осуществляющие маркетинг, менеджмент производства, научно-технического прогресса и т.п.;

б) исходя из характера общих функций, выполняемых в процессе управления, формируются органы, занимающиеся планированием, организующие производство, труд и управление, контролирующие все процессы в организации.

Отношения между элементами структуры управления поддерживаются благодаря связям, которые принято подразделять на горизонтальные и вертикальные связи. Первые носят характер согласования и являются одноуровневыми. Вторые - это отношения подчинения. Необходимость в них возникает при иерархичности построения системы управления, то есть при наличии различных уровней управления, на каждом из которых преследуются свои цели.

При двухуровневой структуре создаются верхние звенья управления (руководство организацией в целом) и низовые звенья (менеджеры, непосредственно руководящие работой исполнителей). При трех и более уровнях в ОСУ формируется так называемый средний слой, который в свою очередь может состоять из нескольких уровней.

В структуре управления организацией различаются линейные и функциональные связи. Первые суть отношения по поводу принятия и реализации управленческих решений и движения информации между так называемыми линейными руководителями, то есть лицами, полностью отвечающими за деятельность организации и ее структурных подразделений. Функциональные связи сопрягаются с теми или иными функциями менеджмента. Соответственно используется такое понятие, как полномочия: линейного персонала, штабного персонала и функциональные. Полномочия линейных руководителей дают право решать все вопросы развития вверенных им организаций и подразделений, а также отдавать распоряжения, обязательные для выполнения другими членами организации (подразделений). Полномочия штабного персонала ограничиваются правом планировать, рекомендовать, советовать или помогать, но не приказывать другим членам организации выполнять их распоряжения.

Между всеми выше составляющими ОСУ существуют сложные отношения взаимозависимости: изменения в каждой из них (числа элементов и уровней, количества и характера связей и полномочий работников) вызывают необходимость пересмотра всех остальных. Так, если руководством организации принято решение о введении в ОСУ нового органа, например, отдела маркетинга (функции которого ранее никто не выполнял), нужно одновременно дать ответ на следующие вопросы: какие задачи будет решать новый отдел, кому он будет непосредственно подчинен, какие органы и подразделения организации будут доводить до него необходимую информацию, на каких иерархических уровнях будет представлена новая служба, какими полномочиями наделяются работники нового отдела, какие формы связей должны быть установлены между новым отделом и другими отделами.

Увеличение количества элементов и уровней в ОСУ неизбежно приводит к многократному росту числа и сложности связей, возникающих в процессе принятия управленческих решений. Следствием этого нередко является замедление процесса управления, что в современных условиях тождественно ухудшению качества функционирования менеджмента организации.

На всех уровнях управления уделяют огромное внимание принципам и методам формирования структур, выбору типа или комбинации видов структур. Многосторонность содержания структур управления предопределяет множество принципов их формирования. Главные из этих принципов могут быть сформулированы следующим образом [11]:

1. Организационная структура управления должна, прежде всего, отражать цели и задачи организации, а, следовательно, быть подчиненной производству и его потребностям.

2. Следует предусматривать оптимальное разделение труда между органами управления и отдельными работниками, обеспечивающее творческий характер работы и нормальную нагрузку, а также надлежащую специализацию.

3. Формирование структуры управления надлежит связывать с определением полномочий и ответственности каждого работника и органа управления, с установлением системы вертикальных и горизонтальных связей между ними.

4. Между функциями и обязанностями, с одной стороны, и полномочиями и ответственностью с другой, необходимо поддерживать соответствие, нарушение которого приводит к дисфункции системы управления в целом.

5. Организационная структура управления призвана быть адекватной в социально-культурной среде организации, оказывающей существенное влияние на решения, относительно уровня централизации и детализации, распределения полномочий и ответственности, степени самостоятельности и масштабов контроля руководителей и менеджеров.

Реализация этих принципов означает необходимость учета при формировании (или перестройке) структуры управления множества различных факторов воздействия на ОСУ.

Главный фактор, «задающий» возможные контуры и параметры структуры управления - сама организация. Известно, что организации различаются по многим критериям. Большое разнообразие организаций в Российской Федерации предопределяет множественность подходов к построению управленческих структур. Подходы эти различны в организациях коммерческих и некоммерческих, крупных, средних и малых, находящихся на разных стадиях жизненного цикла, имеющих разный уровень разделения и специализации труда, его кооперирования и автоматизации, иерархических и «плоских», и так далее. Очевидно, что структура управления крупными предприятиями более сложна по сравнению с той, какая нужна небольшой фирме, где все функции менеджмента подчас сосредотачиваются в руках одного - двух членов организации (обычно руководителя и бухгалтера), где соответственно нет необходимости проектировать формальные структурные параметры. По мере роста организации, а значит, и объема управленческих работ, развивается разделение труда, и формируются специализированные звенья (например, по управлению персоналом, производством, финансами, инновациями и т.п.), слаженная работа которых требует координации и контроля.

Важно обратить внимание на сопряжение структуры управления с фазами жизненного цикла организации. На стадии зарождения организации управление нередко осуществляется самим предпринимателем. На стадии роста происходит функциональное разделение труда менеджеров. На стадии зрелости в структуре управления чаще всего реализуется тенденция к децентрализации. На стадии спада обычно разрабатываются меры по совершенствованию управленческой структуры в соответствии с потребностями и тенденциями в изменении производства. Наконец, на стадии прекращения существования организации структура управления или полностью разрушается, или происходит ее реорганизация.

На формирование структуры управления оказывают влияние изменения организационных форм, в которых функционируют предприятия. Так, при вхождении фирмы в состав какого-либо объединения, (концерна, ассоциации и т.п.) происходит перераспределение управленческих функций (часть функций централизуется), поэтому меняется и структура управления фирмы. Если предприятие остается самостоятельным и независимым, но становится частью сетевой организации, объединяющей на временной основе ряд взаимосвязанных предприятий (чаще всего для использования благоприятной ситуации), ему приходится вносить в свою управленческую структуру ряд изменений. Это связано с необходимостью усиления функций координации и адаптации к системам менеджмента других компаний, входящих в сеть.

Важный фактор формирования управленческих структур - уровень развития на предприятии информационной технологии. Общая тенденция к децентрализации «электронного интеллекта», то есть к росту числа персональных компьютеров при одновременном расширении использования на уровне предприятия локальных сетей, ведет к ликвидации или сокращению объема работ по ряду функций на среднем и низовом уровне. Это относится, прежде всего, к координации работы подчиненных звеньев, передаче информации, обобщению результатов деятельности отдельных сотрудников. Прямым результатом использования локальных сетей может быть расширение сферы контроля руководителей при сокращении числа уровней управления на предприятии.

Большинство организаций непрерывно совершенствуют свои организационные структуры. В настоящее время характерно сокращение числа управленческих уровней: переход от 8-12 управленческих уровней к 4-5, т.е. сокращение числа промежуточных звеньев между главным управляющим и сотрудниками.

## 1.3 Методики оценки организационных структур

Информационный подход к оценке управленческих структур первоначально был ориентирован на отображение и анализ пространственно распределенных систем и опирался на аппарат математической теории поля, поэтому он получил название теории информационного поля. В дальнейшем на основе этой теории был получен вариант информационного описания объектов с сосредоточенными параметрами (то есть с выделением дискретных элементов), что более удобно для исследования экономических объектов и процессов.

Можно выделить три этапа отражения действительности, из которых два пассивных - этап чувственного и этап логического отражения, а один активный - этап прагматического отражения. Поскольку единственным продуктом всякого отражения является информация, то продуктами вышеназванных этапов отражения являются чувственная, логическая и прагматическая информации.

Чувственная информация J вводится как мера отраженной в нашем сознании элементной базы системы в формуле

J = , (1)

где А - общее количество каких-либо знаков, воспринимаемых измерительными приборами или нашими органами чувств; ΔА - «квант», с точностью до которого нас интересует воспринимаемая информация, или разрешающая способность прибора.

Логическая информация (сущность) Н, в отличие от J, всегда относящейся к конкретным объектам или свойствам, характеризует целый класс однородных в определенном отношениии объектов или свойств, являясь семантическим синтезом законов логики, правил функционирования системы и ее элементов, образующих функционал ее существования.

Согласно основному закону классической логики Аристотеля собственная сущность (суть) системы обратно пропорциональна объему понятия о ней, т. е.

. (2)

Логическую информацию H можно определить не только через параметры синтезирующей ее системы (человека, автоматизированной информационной системы). Если учесть, что H характеризует не единичный объект, а класс однородных в определенном смысле объектов или свойств, то это значение можно определить через интеграл плотности вероятности f(Ji), полагая, что J имеет значение Ji

. (3)

В частном случае вместо плотности вероятности можно ввести класс однородных объектов с вероятностью qi представив Ji в логарифмической форме; тогда получим

. (4)

Значения qi и pi могут быть не равны, но возможны ситуации (как, например, в случае передачи сигналов по каналам связи, когда qi = pi), что имеет место в формуле Шеннона

. (5)

Прагматическая информация Hц описывается аналогичной моделью, только под J понимается информация о средствах достижения цели, а под n - количество бит информации о средствах на бит информации о цели (результате).

Семантическая информация Н так же, как и прагматическая Hц, может иметь статистическую трактовку, т. е. Нц можно определять аналогично (2.5), только в этом случае для практических приложений часто удобнее заменить вероятность недостижения цели pi на сопряженную (’):

 (6)

где  - вероятность достижения цели; qi - вероятность того, что оцениваемая компонента будет использована для достижения цели.

Наряду с рассмотренными основными понятиями информационного подхода, в [27] вводится еще характеристика сложности содержания, - в случае семантической информации, или смысла - в случае прагматической информации.

Сложность (содержание, смысл) С определяется пересечением (в частных случаях - произведением) J и H

. (7)

Соотношение (2.8) получено в теории информационного поля, и доказательство его требует введения ряда понятий этой теории. Поэтому ограничимся таким пояснением: H характеризует содержание (суть) только единицы чувственной информации, а для того чтобы охарактеризовать сложность всей информации, нужно, естественно, умножить H на количество чувственной информации J. Для случая прагматической информации сложность Сц (смысл информации для достижения поставленной цели) должна определяться с учетом прагматической чувственной информации, влияющей на достижение цели, и прагматической сути Hц .

Для систем организационного управления интерпретация С зависит от конкретных условий его применения. В частности, в зависимости от того, применительно к характеристике всей системы или ее элементов используется С, можно говорить о системной, собственной и взаимной сложности.

Если учесть чувственную информацию I, то получим соотношение, определяющее взаимосвязь системной СС, собственной СО и взаимной СВ сложности системы, т. е.

СС =Со+СВ. (8)

Собственная сложность Со характеризует суммарную сложность (содержание) элементов системы вне связи их между собой (в случае прагматической информации - суммарная сложность элементов, влияющих на достижение цели). Системная сложность СС отражает содержание системы как целого (например, сложность ее использования). Наконец, СВ характеризует степень взаимосвязи элементов в системе (т. е. сложность ее устройства, схемы, структуры).

Обратим внимание на тот факт, что суммарная собственная сложность элементов в устойчивых системах больше, чем системная, т. е.   
Со > СС. Большим, нежели СС, может быть и СВ. Например, сложность телевизора СС для пользователя меньше сложности его конструкции СВ и суммарной сложности СС (возможностей) элементов, из которых собран телевизор. Так что иногда бытующее выражение «целое больше своих частей» не следует понимать буквально. Количественно содержание целого может быть меньше, но качественно его свойства принципиально новы по сравнению со свойствами составляющих его частей.

В изображениях иерархической структуры системы способ вычленения элементов не определен, и ее «читать» можно неодинаково.

Так, элементами можно считать каждую ветвь иерархической структуры, полагая, что она имеет два возможных состояния («участвует» или «не участвует» в принятии решения), т. е. оценивается минимальной единицей чувственной информации J = 1 бит. А можно разделить структуры на элементы так, как показано на рис. 2.2 а, б, которые являются основой для структур на рисунке 2.2 в, г, д, е, и тогда каждый элемент по отношению к системе может оцениваться так же, как участвующий или нет в принятии решения (т. е. его J = 1 бит), но с разными «способностями», которые оцениваются числом его возможных состояний, отражаемых в оценке его Н: для элементов с двумя состояниями (примем для упрощения состояния равновероятными) H=1 бит; для элементов с четырьмя равновероятными состояниями Н = 2 бита и т. д.

. (9)

Наименьшая централизация характерна для структуры на рисунке 2.2 е. У подобных, сильно неравномерных структур, когда одной из вершин подчинено существенно большее число составляющих, чем другой, есть существенный недостаток: малая разница в оценках (которые в данном случае удобно трактовать как потенциал, значимость, характеризующие влияние соответствующей вершины на принятие решений) для системы в целом (3 бита) и для вершины, которой подчинено шесть составляющих (2,7 бита), приводит к тому, что ЛПР, находящиеся в последней, начинают вести себя практически независимо от руководителя системы. Этот недостаток довольно часто проявлялся на практике, но его иногда пытались объяснить квалификацией и авторитетом соответствующих ЛПР, а проведенный анализ показывает, что это характеристика структуры, а не частного лица.

Оценки J, Н, а, соответственно и С, можно не переводить в привычные для измерения информации единицы - биты, т. е. не прибегать к логарифмической шкале представления оценок, как это показано, например, в работе [23]. При этом получаются более громоздкие вычисления и несколько иная степень различия Н и С (логарифм сжимает шкалу оценок), но характер выводов сохраняется.

Наряду с рассмотренными оценками, характеризующими структурные особенности систем, взаимоотношения частей и целого, часто бывает полезно оценить систему и ее структуру с точки зрения затрат труда на принятие решения в процессе функционирования системы. В частности, большинство управленческих решений связано с выбором исполнителя или адресата из числа сотрудников, подчиненных той или иной вершине оргструктуры. Так, определяя, до сведения каких из подразделений или отдельных сотрудников нужно довести соответствующую директивную или отчетную информацию, ЛПР затрачивает труд на прочтение хотя бы заголовков распределяемых документов, на сопоставление их с наименованиями подчиненных подразделений (или с темами, выполняемыми подчиненными ему сотрудниками), т. е. на переработку определенной информации (которую можно оценить в буквах, словах, абзацах).

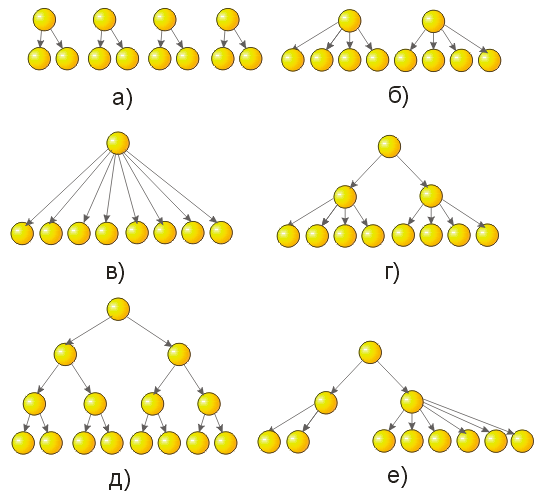


Рисунок 1 – Иерархические структуры систем

Таким образом, реальные затраты управленческого труда, то есть фактический смысл задачи Сф, превосходят ее структурный смысл во столько раз, во сколько раз фактически перерабатываемая для принятия решения информация Jф превосходит структурную J

Сф=, (10)

где Нi -cущность (потенциал) i-гo элемента структуры; Jф- информация того же элемента (которая определяется с учетом числа состояний элемента, то есть числа подчиненных ему составляющих).

Суммарная оценка затрат труда на принятие решений (выбор) при прохождении по структуре сверху вниз важна не только при оценке оргструктур, но и при определении структур банков данных, алгоритмов поиска информации в них.

Для оценки систем можно также использовать характеристику полезной производительности (информационной мощности) N. Можно ввести коэффициент полезного действия γ структуры, определяя его с учетом полных возможностей структуры C∑., и используемых ее возможностей С с точки зрения данной цели :

γ = C/C∑. (11)

Информационный подход к анализу систем имеет широкий спектр приложений. Наряду с рассмотренными оценками структур и процессов принятия решений, что имеет особо важное значение для экономических объектов, подход позволяет получать объединение разнородных критериев при решении многокритериальных задач, проводить сравнительный анализ влияния разнородных нововведений на реализацию целей систем управления, оценивать переходные процессы принятия решений, тенденции развития систем различной физической природы, принципиальные возможности реализации системы как самоорганизующейся, развивающейся, переход ее на новый качественный уровень.

Рассмотренные выше методики и средства системного анализа особенно необходимы в том случае, когда предстоит кардинальная переработка существующей системы управления, или, когда нужно создавать заново структуру системы управления в связи с полностью изменившимися целевыми установками.

# РАЗРАБОТКА ТРЕБОВАНИЙ К ПО ПРОЦЕДУРЫ ОЦЕНКИ ОРГСТРУКТУР

## 2.1 Разработка функциональной модели

В ходе разработки были разработаны следующие модели требований к ПО для оценки оргструктуры предприятия. Основными требованиями стали гибкость и возможность критериальной оценки.



Рисунок 12 - Диаграмма прецедентов

C:\Users\student\Downloads\Untitled Diagram (1).png

ПО

Рисунок 13 - Диаграмма последовательности

## 2.2 Выбор среды реализации ПО

Для разработки приложения была выбрана среда Visual Studio 2015 – это передовая среда разработки на языке C#, предназначенная для создания интерактивных приложений с пользовательским интерфейсом для настольных компьютеров, рабочих станций, сенсорных дисплеев, информационных терминалов и Интернета. Миллионы разработчиков по всему миру выбирают Visual Studio, поскольку на сегодняшний день это единственная действительно двусторонняя визуальная среда с широкими функциональными возможностями для быстрой разработки на языке C# приложений с развитым пользовательским интерфейсом и приложений баз данных. Приложения, созданные в среде Visual Studio, отличаются высочайшим быстродействием и компактностью, а также обладают полнофункциональным пользовательским интерфейсом и способностью подключаться практически к любой базе данных или другому источнику данных, не требуя дополнительной доработки.

Visual Studio включает в себя мощный интегрированный набор инструментов визуальную двустороннюю среду для создания интерактивных приложений с пользовательским интерфейсом и расширяемую платформу визуальных компонентов, содержащую свыше 250 стандартных классов и компонентов. Многочисленные сторонние поставщики и сообщество Visual Studio, насчитывающее в своих рядах более 1,5 миллиона разработчиков, предлагают тысячи дополнительных компонентов Visual Studio практически для каждого конкретного приложения, технологии, функции или задачи.

# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО ОЦЕНКЕ ОРГСТРУКТУР

## Описание процесса разработки

В процессе разработки была создана система для построения оргструктур. Основным хранилищем данных являются списки объектов класса такие как:

List<TextBox> save\_textbox1;

public static List<double> compare1;

Они выполняют роль контейнеров для всех элементов оргструктуры.

Так как приложение имеет четыре уровня детализации, индексами каждого уровня, а так же счетчиками количества элементов каждого уровная являются переменные i, i1,i2,i3. Создание элементов формы происходит по нажатию на клавишу создать элемент (Рисунок 4), обработчик события которой создает TextBox на форме, пример такого обработчика приведен ниже:

private void button1\_Click(object sender, EventArgs e)

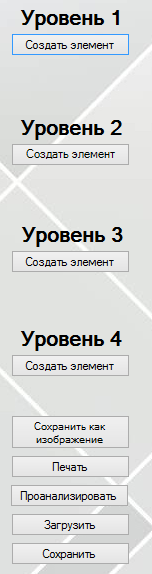


Рисунок 14 – Клавиши построения

После построения элементов происходит построение связей. Флагом для включения режима построения связей является checkbox1. При активации флага становится доступным построение связей, нажав на элемент и по двойному щелчку по нижестоящему элементу структуры.

При добавлении связи она заносится в контейнер связей, который хранит данные об объектах начала и конца связи.

public void line\_DoubleClick(object sender, EventArgs e)  
Для обеспечения удобства пользования приложением установлены инкрементируемые переменные-счетчики положения элементов формы: х, х1, х2, х3, y, y1,y2,y3.

Для сохранения результатов построения оргструктуры использована СУБД MS Access. БД хранит координаты и содержание элементов оргструктуры. Схема данных приведена на рисунке 5.

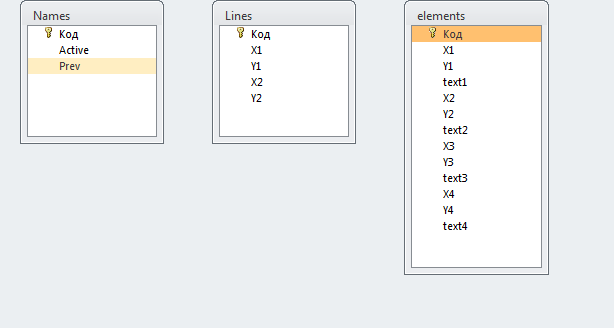


Рисунок 15 – Схема данных

## Инструкция пользователя

Для начала работы в среде разработки организационных структур необходимо ввести название вашей организации в поле ввода вверху экрана:

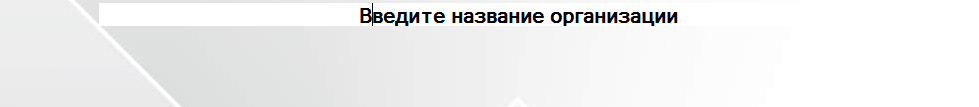
****

Рисунок 16 – Ввод названия организации

Для построения организационной структуры введите управляющий элемент, нажав на кнопку «Создать элемент» под первым уровнем структуры и введите его наименование в появившееся поля ввода (Рисунок 6).

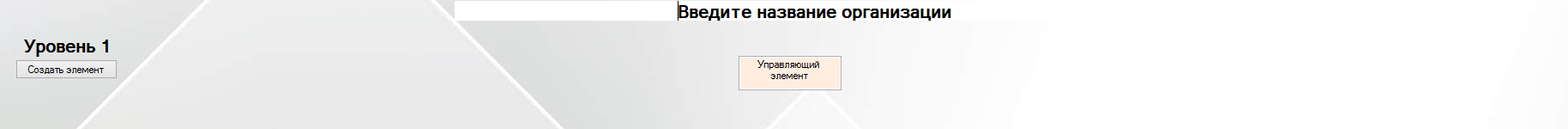
****

Рисунок 17 – Ввод главного управляющего элемента

Продолжить добавление элементов других уровней возможно с помощью аналогичных кнопок соответственно уровню оргструктуры.

Для создания связей между элементами необходимо отметить галочкой пункт «Установить связи» после чего нажать ЛКМ по управляющему элементу, а затем дважды на подчиненном. Результат построения оргструктуры приведен на рисунке.

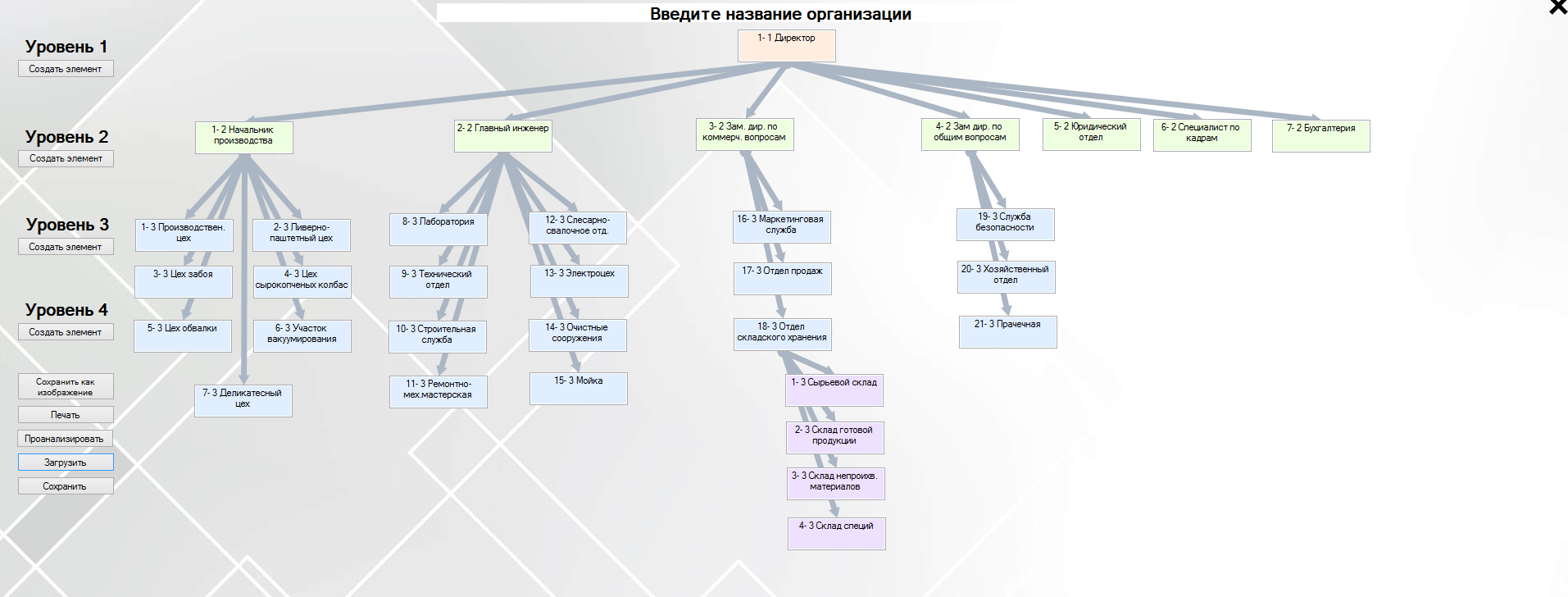
****

Рисунок 18 – Внешний вид построенной структуры

При необходимости построения прочих вариантов структуры нажмите на клавишу «+». На рисунках 9, 10 и 11 показаны возможные варианты построения оргструктуры ОАО “Мясоперерабатывающий комплекс Динской”.

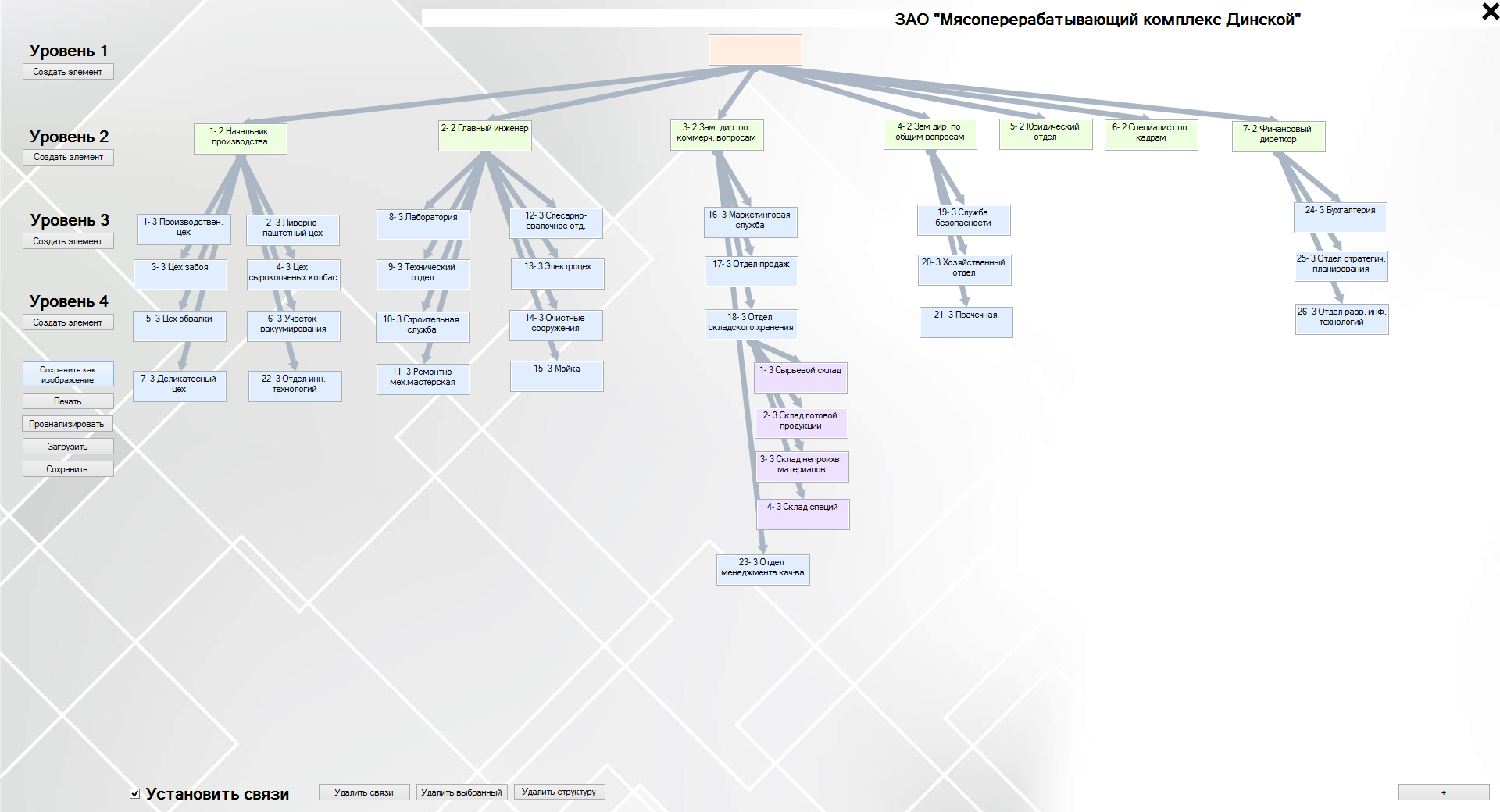


Рисунок 19 – Второй вариант построения оргструктуры

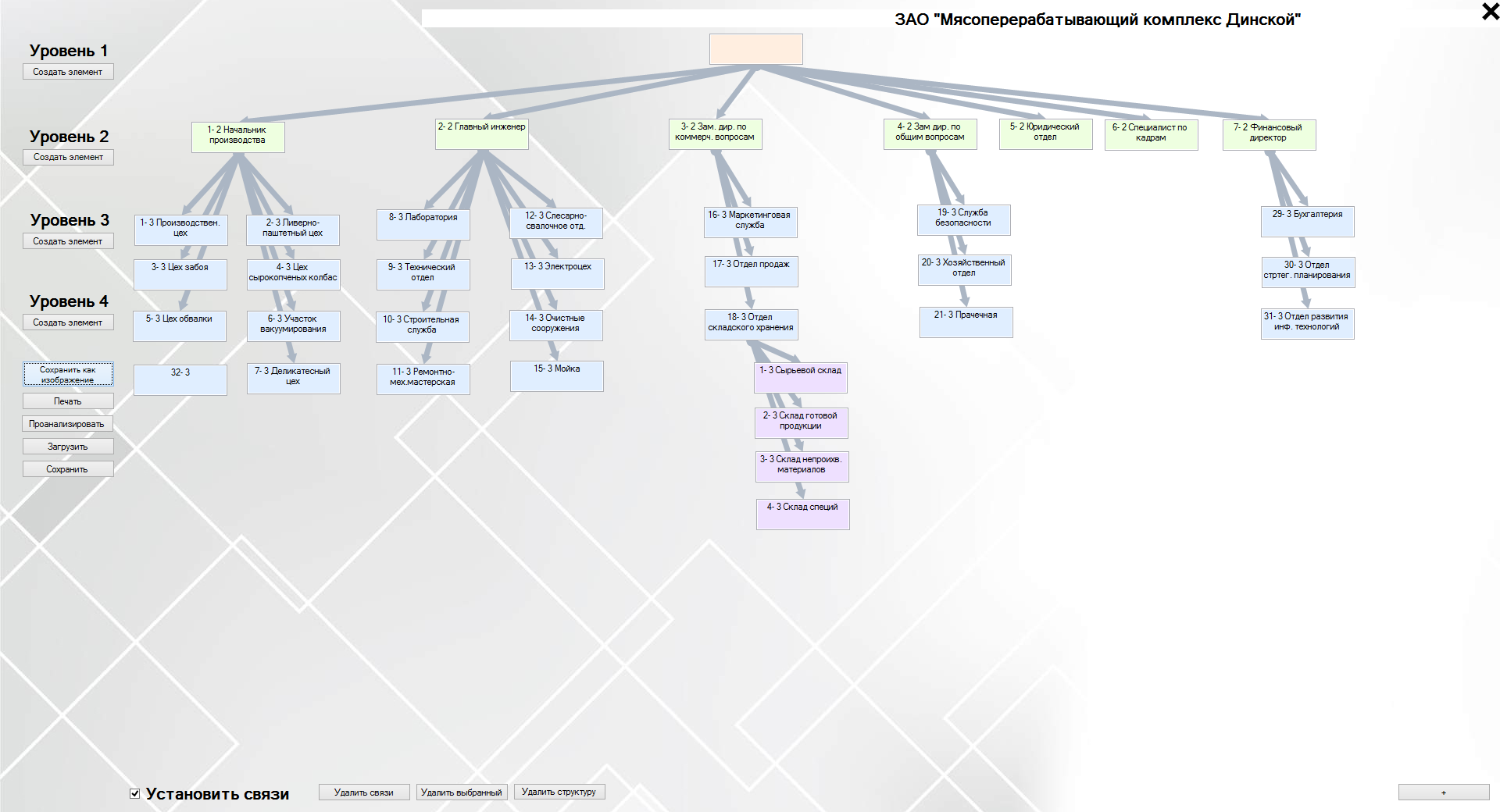


Рисунок 20 – Третий вариант построения оргструктуры

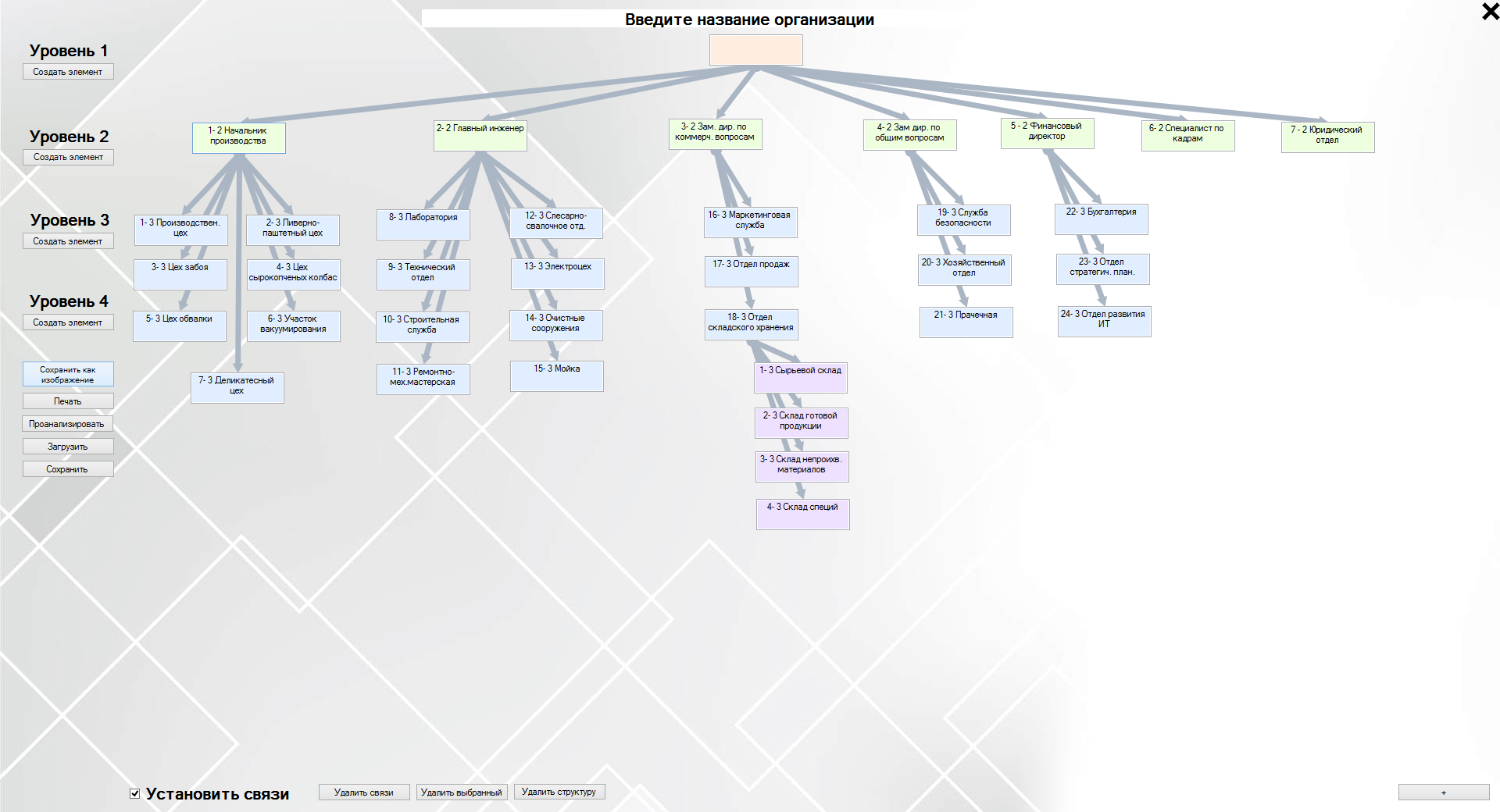


Рисунок 21 – Четвертый вариант построения оргструктуры

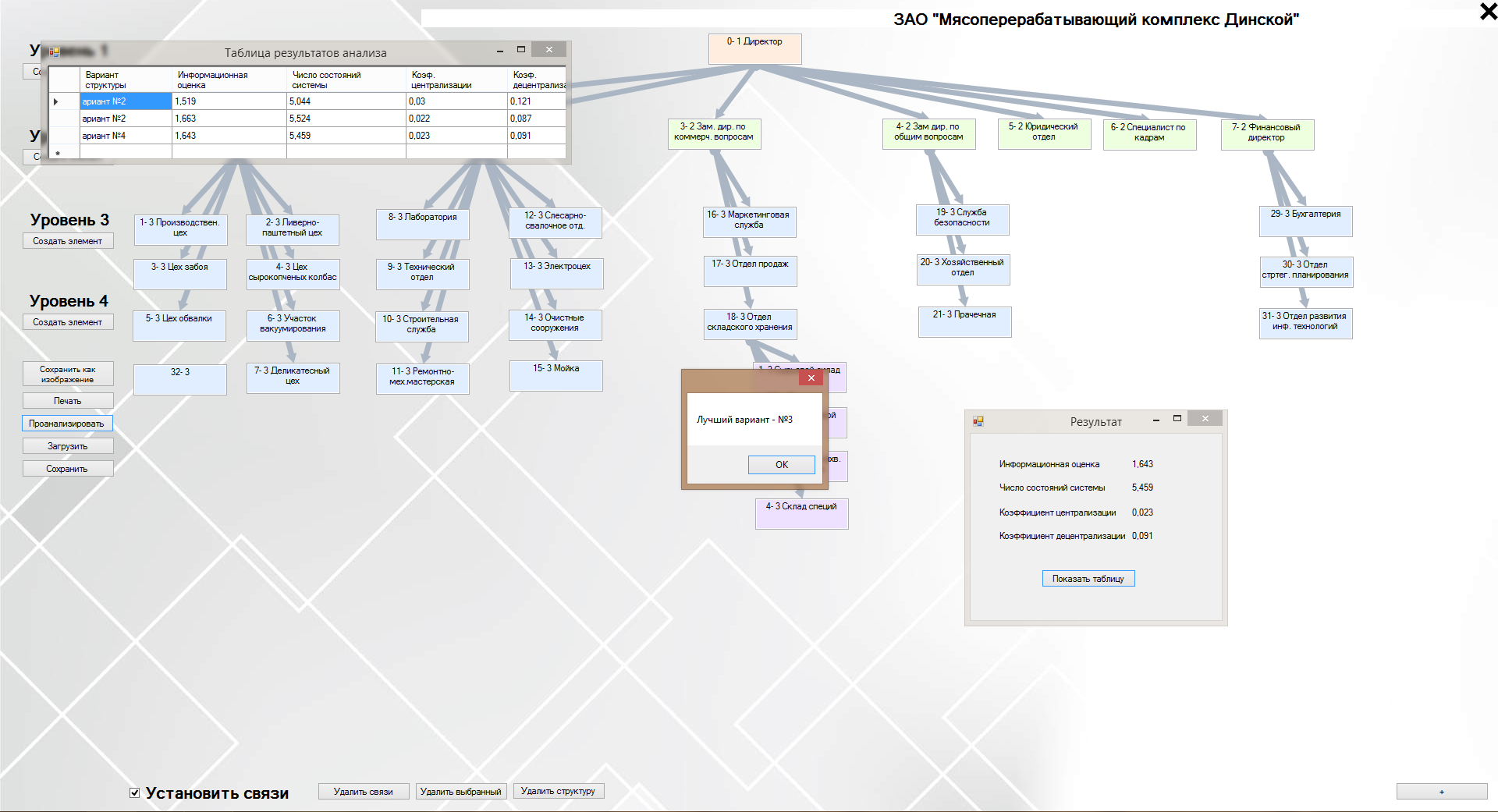


Рисунок 21 – Результирующая таблица и результаты анализа

Для анализа построенной структуры нажмите клавишу «Проанализировать», после чего вы увидите перед собой окно с оценкой Вашей организационной структуры (Рисунок 8).

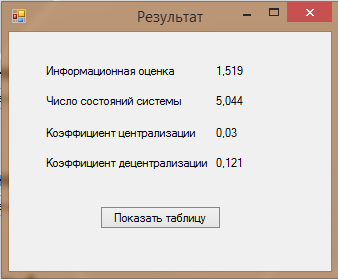
****

Рисунок 22 – Окно результата анализа структуры

Для сбора результатов всех вариантов организационных структур необходимо на каждой построенной диаграмме нажать клавишу «Проанализировать», а затем на последней диаграмме вызвать таблицу результатов. В появившемся окне вы можете сохранить результаты в табличном виде.

Для сохранения текущего состояния структуры нажмите клавишу «Сохранить», аналогично для загрузки сохраненного состояния нажмите клавишу «Загрузить». Для сохранения диаграммы как изображения нажмите клавишу «Сохранить как изображение» (изображения сохраняются в папке Diagrams рядом с приложением). Для печати структуры нажмите клавишу «Печать».

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гриценко Ю.Б. Архитектура предприятия [Электронный ресурс]: учебное пособие/ Гриценко Ю.Б.— Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, — 264 c.— Режим доступа: http://www.iprbookshop.ru/14005.— ЭБС «IPRbooks»
2. Абызова Е. В. Методика формирования организационной структуры системы управления Краснодарским краевым союзом потребительских обществ и ее когнитивное моделирование // Научный журнал Куб ГАУ [Электронный ресурс]. – Краснодар: Куб ГАУ, 2015. - № .- Режим доступа: http: //eg. Kubagro. ru/ 2015/ .- 0,9 п. л.
3. Абызова Е. В. Дерево целей и функций системы управления региональной потребительской кооперацией / Абызова Е.В., Першакова Т.В., Дейнека А.В., // Научный журнал Куб ГАУ [Электронный ресурс]. – Краснодар: Куб ГАУ, 2005. - № .- Режим доступа: http: //eg. Kubagro. ru/ 2015/.- 0,7 п. л.(авт. 0,5)
4. Абызова Е. В. Информационный анализ организационных структур систем управления региональной потребительской кооперацией / Абызова Е.В., Першакова Т.В., Дейнека А.В., // Научный журнал Куб ГАУ [Электронный ресурс]. – Краснодар: Куб ГАУ, 2005. - № .- Режим доступа: http: //eg. Kubagro. ru/ 2015/.- 0,8 п. л. (авт. 0,6).
5. Акофф Р. Планирование будущего корпорации. –М.: Мир, 1985. –185с.
6. Бабешко Л.О. Коллокационные модели прогнозирования в финансовой сфере.- М.: «Экзамен», 2014г. – 288с.
7. Барановская Т.П. Системный анализ организационной структуры управления автодорожной отраслью региона /  Т.П. Барановская, О.К. Безродный, В.И. Лойко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2004. – №06(008). С. 168 – 201. – IDA [article ID]: 0080406018. – Режим доступа: http://ej.kubagro.ru/2004/06/pdf/18.pdf, 2,125 у.п.л.

# ПРИЛОЖЕНИЕ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Printing;

using System.Drawing.Imaging;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Data.SqlClient;

using System.Data.OleDb;

using System.Runtime.Serialization.Formatters.Binary;

using System.Xml.Serialization;

using Microsoft.Office.Core;

namespace Systanalis

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

this.SetBounds(0, 0, Screen.PrimaryScreen.WorkingArea.Width, Screen.PrimaryScreen.WorkingArea.Height);

button6.Location = new Point(button6.Location.X , this.Height - 37);

button5.Location = new Point(button5.Location.X, this.Height - 37);

button8.Location = new Point(button8.Location.X,this.Height - 37);

checkBox1.Location = new Point(checkBox1.Location.X,this.Height - 37);

button15.Location = new Point(button15.Location.X, this.Height - 37);

button11.Name = "Cancel";

textBox1.Name = "Organization";

textBox1.Location = new Point(this.Width / 2 - textBox1.Width / 2, 12);

oleDbConnection1.ConnectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source =" + System.IO.Path.Combine(Application.StartupPath, "data.accdb");

;

oleDbConnection2.ConnectionString = "Provider = Microsoft.ACE.OLEDB.12.0; Data Source =" + System.IO.Path.Combine(Application.StartupPath, "data.accdb");

oleDbConnection1.Open();

oleDbConnection2.Open();

this.BackgroundImageLayout = ImageLayout.None;

this.printDocument1.PrintPage += new PrintPageEventHandler(printDocument1\_PrintPage);

this.Name = "Вариант №" + (formcount+1);

this.Text = "Вариант №" + (formcount+1);

this.Activate();

if (this.Name == "Вариант №1")

forms.Add(this);

foreach (Control qew in this.Controls)

DoubleBuffered = true;

textBox1.TextAlign = HorizontalAlignment.Center;

}

bool isDown;

static public string FileName;

static public Form3 tabl = new Form3();

static public List<Form> forms = new List<Form>();

static public List<Form> forms2 = new List<Form>();

static public int formcount = 0;

int i = 0, count = 0, drawcount = 0;

int x = 170, y = 60;

int i1 = 0;

int x1 = 170, y1 = 160;

int i2 = 0;

int x2 = 170, y2 = 260;

int i3 = 0;

int x3 = 170, y3 = 360;

str[] linee = new str[100];

string name;

Control draw = null, next = null;

List<TextBox> save\_textbox1 = new List<TextBox>();

List<TextBox> save\_textbox2 = new List<TextBox>();

List<TextBox> save\_textbox3 = new List<TextBox>();

List<TextBox> save\_textbox4 = new List<TextBox>();

public static List<double> compare1 = new List<double>();

public static List<double> compare2 = new List<double>();

public static List<double> compare3 = new List<double>();

public static List<double> compare4 = new List<double>();

//обработчики уровней

//1

private void button1\_Click(object sender, EventArgs e)

{

TextBox str = new TextBox();

str.Multiline = true;

str.Name = "Элемент управления" + i + "-1";

str.SetBounds(x, y, 120, 40);

str.TextAlign = HorizontalAlignment.Center;

str.Location = new Point(str.Location.X, str.Location.Y);

Color color = Color.FromArgb(255, 238, 224) ;

str.BackColor = color;

this.Controls.Add(str);

i++;

x += str.Width + 15;

str.Text = i.ToString() + "- 1";

if ((double)i % 7 == 0)

{

y += 45;

x = 170;

y1 += 45;

y2 += 45;

y3 += 45;

}

str.MouseDown += str\_MouseDown;

str.MouseMove += str\_MouseMove;

str.MouseUp += str\_MouseUp;

str.DoubleClick += line\_DoubleClick;

str.Click += save\_name;

save\_textbox1.Add(str);

/\*DataTable el = this.mydataset21.elements;

DataTable ind = this.mydataset2.indexes;

DataTable names = this.namess1.Names;

DataTable linees = this.lines11.Lines;

oleDbDataAdapter1.Fill(el);

oleDbDataAdapter2.Fill(ind);

oleDbDataAdapter3.Fill(linees);

oleDbDataAdapter4.Fill(names);\*/

}

//2

private void button2\_Click(object sender, EventArgs e)

{

TextBox str1 = new TextBox();

str1.Multiline = true;

str1.Name = "Элемент управления" + i1 + "-2";

str1.SetBounds(x1, y1, 120, 40);

Color color1 = Color.FromArgb(238, 255, 224);

str1.BackColor = color1;

this.Controls.Add(str1);

str1.TextAlign = HorizontalAlignment.Center;

i1++;

x1 += str1.Width + 15;

str1.Text = i1.ToString() + "- 2";

if ((double)i1%7 == 0)

{

y1 += 45;

x1 = 170;

y2 += 45;

y3 += 45;

}

str1.MouseDown += str\_MouseDown;

str1.MouseMove += str\_MouseMove;

str1.MouseUp += str\_MouseUp;

str1.DoubleClick += line\_DoubleClick;

str1.Click += save\_name;

save\_textbox2.Add(str1);

}

//3

private void button3\_Click(object sender, EventArgs e)

{

TextBox str2 = new TextBox();

str2.Multiline = true;

str2.Name = "Элемент управления" + i2 + "-3";

str2.SetBounds(x2, y2, 120, 40);

Color color2 = Color.FromArgb(224, 238, 255);

str2.BackColor = color2;

this.Controls.Add(str2);

str2.TextAlign = HorizontalAlignment.Center;

i2++;

x2 += str2.Width + 15;

str2.Text = i2.ToString() + "- 3";

if ((double)i2 % 7 == 0)

{

x2 = 170;

y2 += 45;

y3 += 45;

}

str2.MouseDown += str\_MouseDown;

str2.MouseMove += str\_MouseMove;

str2.MouseUp += str\_MouseUp;

str2.DoubleClick += line\_DoubleClick;

str2.Click += save\_name;

save\_textbox3.Add(str2);

}

//4

private void button4\_Click(object sender, EventArgs e)

{

TextBox str3 = new TextBox();

str3.Multiline = true;

str3.Name = "Элемент управления" + i3 + "-4";

str3.SetBounds(x3, y3, 120, 40);

Color color3 = Color.FromArgb(238, 224, 255);

str3.BackColor = color3;

this.Controls.Add(str3);

str3.TextAlign = HorizontalAlignment.Center;

i3++;

x3 += str3.Width + 15;

str3.Text = i3.ToString() + "- 3";

if ((double)i3 % 7 == 0)

{

x3 = 170;

y3 += 45;

}

str3.MouseDown += str\_MouseDown;

str3.MouseMove += str\_MouseMove;

str3.MouseUp += str\_MouseUp;

str3.DoubleClick += line\_DoubleClick;

str3.Click += save\_name;

save\_textbox4.Add(str3); ;

}

private void button7\_Click(object sender, EventArgs e)

{

//анализ

if (i > 0 && i1 > 0 && i2 > 0 || i3 > 0)

{

Form2 sysanalis = new Form2();

sysanalis.Name = "Вариант №" + formcount;

sysanalis.Activate();

forms2.Add(sysanalis);

sysanalis.Show();

sysanalis.label3.Text = Math.Round(Math.Log10(i + i1 + i2 + i3), 3).ToString();

sysanalis.label4.Text = Math.Round(Math.Log(i + i1 + i2 + i3, 2), 3).ToString();

sysanalis.label6.Text = Math.Round(((double)i / (i + i1 + i2 + i3)), 3).ToString();

sysanalis.Width = Convert.ToInt32(sysanalis.label3.Location.X) + sysanalis.label3.Width + 100;

if (i3 != 0)

{

sysanalis.label8.Text = Math.Round(((double)i3 / (i + i1 + i2 + i3)), 3).ToString();

}

else

{

if (i2 != 0)

{

sysanalis.label8.Text = Math.Round(((double)i2 / (i + i1 + i2)), 3).ToString();

}

else if (i1 == 0 || i == 0)

{

MessageBox.Show("Структура не найдена!");

}

else

{

sysanalis.label8.Text = Math.Round(((double)i1 / (i + i1)), 3).ToString();

}

}

tabl.dataGridView1.ColumnCount = 5;

tabl.dataGridView1.ColumnHeadersVisible = true;

tabl.dataGridView1.Columns[0].Name = "Вариант структуры";

tabl.dataGridView1.Columns[1].Name = "Информационная оценка";

tabl.dataGridView1.Columns[2].Name = "Число состояний системы";

tabl.dataGridView1.Columns[3].Name = "Коэф. централизации";

tabl.dataGridView1.Columns[4].Name = "Коэф. децентрализации";

DataGridViewRow nrow = new DataGridViewRow();

DataGridViewCell r = new DataGridViewTextBoxCell();

DataGridViewCell r1 = new DataGridViewTextBoxCell();

DataGridViewCell r2 = new DataGridViewTextBoxCell();

DataGridViewCell r3 = new DataGridViewTextBoxCell();

DataGridViewCell r4 = new DataGridViewTextBoxCell();

DataGridViewCell r5 = new DataGridViewTextBoxCell();

r.Value = "Вариант №" + forms.Count;

r1.Value = sysanalis.label3.Text;

r2.Value = sysanalis.label4.Text;

r3.Value = sysanalis.label6.Text;

r4.Value = sysanalis.label8.Text;

compare1.Add(Convert.ToDouble(sysanalis.label3.Text));

compare2.Add(Convert.ToDouble(sysanalis.label4.Text));

compare3.Add(Convert.ToDouble(sysanalis.label6.Text));

compare4.Add(Convert.ToDouble(sysanalis.label8.Text));

nrow.Cells.Add(r);

nrow.Cells.Add(r1);

nrow.Cells.Add(r2);

nrow.Cells.Add(r3);

nrow.Cells.Add(r4);

tabl.dataGridView1.Rows.Add(nrow);

tabl.dataGridView1.Width = tabl.Width;

tabl.Height = tabl.dataGridView1.Rows.Count \* 20 + 79;

tabl.button1.Location = new Point(tabl.Width - tabl.button1.Width, tabl.Height - tabl.button1.Height);

tabl.Сохранить.Location = new Point(0, tabl.Height - tabl.Сохранить.Height);

}

else

{

MessageBox.Show("Структура не найдена! Создайте структуру.");

}

}

private void button8\_Click(object sender, EventArgs e)

{

//+

Form1 q = new Form1();

formcount++;

q.Name = "Вариант №" + formcount+1.ToString();

q.Text = "Вариант №" + formcount+1.ToString();

q.Show();

forms.Add(q);

}

private void str\_MouseUp(object sender, MouseEventArgs e)

{

isDown = false;

\_Offset = Point.Empty;

}

//загрузка

private void button9\_Click(object sender, EventArgs e)

{

oleDbDataAdapter1.Fill(mydataset21);

DataTable st = this.mydataset21.elements;

i = 0;

i1 = 0;

i2 = 0;

i3 = 0;

drawcount = 0;

for (int w =0 ; w < st.Rows.Count; w++)

{

Refresh();

if (Convert.ToInt32(st.Rows[w][1]) != 0 && Convert.ToInt32(st.Rows[w][2]) != 0)

{

//возврат первого уровня

TextBox str = new TextBox();

str.Multiline = true;

str.Name = "Элемент управления" + i + "-1";

str.SetBounds(Convert.ToInt32(st.Rows[w][1]), Convert.ToInt32(st.Rows[w][2]), 120, 40);

str.Location = new Point(str.Location.X, str.Location.Y);

Color color = Color.FromArgb(255, 238, 224);

str.BackColor = color;

this.Controls.Add(str);

str.Text = st.Rows[w][3].ToString();

str.TextAlign = HorizontalAlignment.Center;

i++;

x += str.Width + 15;

str.MouseDown += str\_MouseDown;

str.MouseMove += str\_MouseMove;

str.MouseUp += str\_MouseUp;

str.DoubleClick += line\_DoubleClick;

str.Click += save\_name;

save\_textbox1.Add(str);

}

if (Convert.ToInt32(st.Rows[w][4]) != 0 && Convert.ToInt32(st.Rows[w][5]) != 0)

{

//возврат второго уровня

TextBox str1 = new TextBox();

str1.Multiline = true;

str1.Name = "Элемент управления" + i1 + "-2";

str1.SetBounds(Convert.ToInt32(st.Rows[w][4]), Convert.ToInt32(st.Rows[w][5]), 120, 40);

Color color1 = Color.FromArgb(238, 255, 224);

str1.BackColor = color1;

this.Controls.Add(str1);

str1.Text = st.Rows[w][6].ToString();

str1.TextAlign = HorizontalAlignment.Center;

i1++;

x1 += str1.Width + 15;

str1.MouseDown += str\_MouseDown;

str1.MouseMove += str\_MouseMove;

str1.MouseUp += str\_MouseUp;

str1.DoubleClick += line\_DoubleClick;

str1.Click += save\_name;

save\_textbox2.Add(str1);

}

if (Convert.ToInt32(st.Rows[w][7]) != 0 && Convert.ToInt32(st.Rows[w][8]) != 0)

{

//возврат третьего уровня

TextBox str2 = new TextBox();

str2.Multiline = true;

str2.Name = "Элемент управления" + i2 + "-3";

str2.SetBounds(Convert.ToInt32(st.Rows[w][7]), Convert.ToInt32(st.Rows[w][8]), 120, 40);

Color color2 = Color.FromArgb(224, 238, 255);

str2.BackColor = color2;

this.Controls.Add(str2);

str2.Text = st.Rows[w][9].ToString();

str2.TextAlign = HorizontalAlignment.Center;

i2++;

x2 += str2.Width + 15;

str2.MouseDown += str\_MouseDown;

str2.MouseMove += str\_MouseMove;

str2.MouseUp += str\_MouseUp;

str2.DoubleClick += line\_DoubleClick;

str2.Click += save\_name;

save\_textbox3.Add(str2);

}

if (Convert.ToInt32(st.Rows[w][10]) != 0 && Convert.ToInt32(st.Rows[w][11]) != 0)

{

//возврат четвертого уровня

TextBox str3 = new TextBox();

str3.Multiline = true;

str3.Name = "Элемент управления" + i3 + "-4";

str3.SetBounds(Convert.ToInt32(st.Rows[w][10]), Convert.ToInt32(st.Rows[w][11]), 120, 40);

Color color3 = Color.FromArgb(238, 224, 255);

str3.BackColor = color3;

this.Controls.Add(str3);

str3.Text = st.Rows[w][12].ToString();

str3.TextAlign = HorizontalAlignment.Center;

i3++;

x3 += str3.Width + 15;

str3.MouseDown += str\_MouseDown;

str3.MouseMove += str\_MouseMove;

str3.MouseUp += str\_MouseUp;

str3.DoubleClick += line\_DoubleClick;

str3.Click += save\_name;

save\_textbox4.Add(str3);

}

oleDbDataAdapter3.Fill(lines11);

}

oleDbDataAdapter4.Fill(namess1);

DataTable n = this.namess1.Names;

for (int o = 0; o < n.Rows.Count; o++)

{

linee[o] = new str();

foreach (Control tx in this.Controls)

{

if (tx is TextBox)

{

if (tx.Name == n.Rows[o][1].ToString())

{

linee[o].ActiveBox = (TextBox)tx;

}

if (tx.Name == n.Rows[o][2].ToString())

{

linee[o].PrevBox = (TextBox)tx;

}

}

}

}

while (linee[drawcount] != null)

{

Pen pen = new Pen(Color.FromArgb(171, 182, 196), 7);

Graphics formGraphics = this.CreateGraphics();

pen.StartCap = LineCap.ArrowAnchor;

pen.EndCap = LineCap.RoundAnchor;

formGraphics.SmoothingMode = SmoothingMode.HighQuality;

formGraphics.DrawLine(pen, linee[drawcount].ActiveBox.Location.X + linee[drawcount].ActiveBox.Width / 2, linee[drawcount].ActiveBox.Location.Y, linee[drawcount].PrevBox.Location.X + linee[drawcount].PrevBox.Size.Width / 2, linee[drawcount].PrevBox.Location.Y + linee[drawcount].PrevBox.Height);

pen.Dispose();

formGraphics.Dispose();

drawcount++;

}

}

//сохранение

private void button10\_Click(object sender, EventArgs e)

{

/////////////////////////////////////////

oleDbDataAdapter4.Fill(namess1);

for (int del = 0; del < namess1.Names.Rows.Count; del++)

{

namess1.Names.Rows[del].Delete();

}

this.oleDbDataAdapter4.Update(namess1);

/////////////////////////////////////////

oleDbDataAdapter1.Fill(mydataset21);

for (int del = 0; del < mydataset21.elements.Rows.Count; del++)

{

mydataset21.elements.Rows[del].Delete();

}

this.oleDbDataAdapter1.Update(mydataset21);

/////////////////////////////////////////

oleDbDataAdapter2.Fill(mydataset2);

for (int del = 0; del < mydataset2.indexes.Rows.Count; del++)

{

mydataset2.indexes.Rows[del].Delete();

}

this.oleDbDataAdapter2.Update(mydataset2);

/////////////////////////////////////////

oleDbDataAdapter3.Fill(lines11);

for (int del = 0; del < lines11.Lines.Rows.Count; del++)

{

lines11.Lines.Rows[del].Delete();

}

this.oleDbDataAdapter3.Update(lines11);

/////////////////////////////////////////

//сохраняем индексы

mydataset.indexesRow saved\_rows;

saved\_rows = mydataset2.indexes.NewindexesRow();

saved\_rows.i = i;

saved\_rows.i1 = i1;

saved\_rows.i2 = i2;

saved\_rows.i3 = i3;

this.mydataset2.indexes.Rows.Add(saved\_rows);

this.oleDbDataAdapter2.Update(this.mydataset2.indexes);

int save\_index1 = 0, save\_index2 = 0, save\_index3 = 0, save\_index4 = 0;

//сохраняем элементы

while (save\_index1 < i || save\_index2 < i1 || save\_index3 < i2 || save\_index4 < i3)

{

mydataset2.elementsRow saved\_elems;

saved\_elems = mydataset21.elements.NewelementsRow();

if (save\_index1 < i)

{

saved\_elems.X1 = Convert.ToInt32(save\_textbox1[save\_index1].Location.X);

saved\_elems.Y1 = Convert.ToInt32(save\_textbox1[save\_index1].Location.Y);

saved\_elems.text1 = save\_textbox1[save\_index1 ].Text;

save\_index1 ++;

}

else

{

saved\_elems.X1 = 0;

saved\_elems.Y1 = 0;

saved\_elems.text1 = "";

}

if (save\_index2 < i1)

{

saved\_elems.X2 = Convert.ToInt32(save\_textbox2[save\_index2].Location.X);

saved\_elems.Y2 = Convert.ToInt32(save\_textbox2[save\_index2].Location.Y);

saved\_elems.text2 = save\_textbox2[save\_index2].Text;

save\_index2++;

}

else

{

saved\_elems.X2 = 0;

saved\_elems.Y2 = 0;

saved\_elems.text2 = "";

}

if (save\_index3 < i2)

{

saved\_elems.X3 = Convert.ToInt32(save\_textbox3[save\_index3].Location.X);

saved\_elems.Y3 = Convert.ToInt32(save\_textbox3[save\_index3].Location.Y);

saved\_elems.text3 = save\_textbox3[save\_index3].Text;

save\_index3++;

}

else

{

saved\_elems.X3 = 0;

saved\_elems.Y3 = 0;

saved\_elems.text3 = "";

}

if (save\_index4 < i3)

{

saved\_elems.X4 = Convert.ToInt32(save\_textbox4[save\_index4].Location.X);

saved\_elems.Y4 = Convert.ToInt32(save\_textbox4[save\_index4].Location.Y);

saved\_elems.text4 = save\_textbox4[save\_index4].Text;

save\_index4++;

}

else

{

saved\_elems.X4 = 0;

saved\_elems.Y4 = 0;

saved\_elems.text4 = "";

}

this.mydataset21.elements.Rows.Add(saved\_elems);

this.oleDbDataAdapter1.Update(this.mydataset21.elements);

}

//сохранение координат связей

for (int q = 0; q < linee.Length; q++)

{

if (linee[q] != null)

{

Namess.NamesRow saved\_lines;

saved\_lines = namess1.Names.NewNamesRow();

saved\_lines.Active = linee[q].ActiveBox.Name;

saved\_lines.Prev = linee[q].PrevBox.Name;

this.namess1.Names.Rows.Add(saved\_lines);

this.oleDbDataAdapter4.Update(namess1.Names);

}

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

oleDbConnection1.Close();

}

private void button11\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы уверены, что хотите выйти?", "Выйти?", MessageBoxButtons.YesNo) == DialogResult.Yes)

ActiveForm.Close();

}

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

base.Capture = false;

Message m = Message.Create(base.Handle, 0xa1, new IntPtr(2), IntPtr.Zero);

this.WndProc(ref m);

}

private void textBox1\_DoubleClick(object sender, EventArgs e)

{

textBox1.SelectAll();

}

private void button6\_Click(object sender, EventArgs e)

{

//удаление связей

Refresh();

for (int q = 0; q < linee.Length; q++)

linee[q] = null;

count = 0;

drawcount = 0;

}

public void line\_DoubleClick(object sender, EventArgs e)

{

linee[count] = new str();

linee[count].ActiveBox = (TextBox)ActiveControl;

linee[count].PrevBox = (TextBox)draw;

if (checkBox1.Checked)

{

Pen pen = new Pen(Color.FromArgb(171, 182, 196), 7);

System.Drawing.Graphics formGraphics = this.CreateGraphics();

pen.StartCap = LineCap.ArrowAnchor;

pen.EndCap = LineCap.RoundAnchor;

formGraphics.SmoothingMode = SmoothingMode.HighQuality;

if (draw.GetType() == typeof(TextBox))

{

formGraphics.DrawLine(pen, ActiveControl.Location.X + ActiveControl.Width / 2, ActiveControl.Location.Y, draw.Location.X + draw.Size.Width / 2, draw.Location.Y + draw.Height);

pen.Dispose();

formGraphics.Dispose();

}

}

count++;

}

private void str\_MouseDown(object sender, MouseEventArgs e)

{

isDown = true;

if (e.Button == MouseButtons.Left)

{

\_Offset = new Point(e.X, e.Y);

}

}

private Point \_Offset = Point.Empty;

void str\_MouseMove(object sender, MouseEventArgs e)

{

if (isDown)

{

if (linee != null)

{

Refresh();

drawcount = 0;

while (linee[drawcount] != null)

{

Pen pen = new Pen(Color.FromArgb(171, 182, 196), 7);

System.Drawing.Graphics formGraphics = this.CreateGraphics();

pen.StartCap = LineCap.ArrowAnchor;

pen.EndCap = LineCap.RoundAnchor;

formGraphics.SmoothingMode = SmoothingMode.HighQuality;

formGraphics.DrawLine(pen, linee[drawcount].ActiveBox.Location.X + linee[drawcount].ActiveBox.Width / 2, linee[drawcount].ActiveBox.Location.Y, linee[drawcount].PrevBox.Location.X + linee[drawcount].PrevBox.Size.Width / 2, linee[drawcount].PrevBox.Location.Y + linee[drawcount].PrevBox.Height);

pen.Dispose();

formGraphics.Dispose();

drawcount++;

}

}

if (\_Offset != Point.Empty)

{

Point newlocation = ActiveControl.Location;

newlocation.X += e.X - \_Offset.X;

newlocation.Y += e.Y - \_Offset.Y;

ActiveControl.Location = newlocation;

}

}

}

private void button5\_Click(object sender, EventArgs e)

{

//удаление выбранного

DialogResult result;

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

result = MessageBox.Show("Удалить элемент?", "Message", buttons);

if (result == DialogResult.Yes)

{

Controls.RemoveByKey(name);

foreach (TextBox fnd in save\_textbox1)

{

if (fnd.Name == name)

{

save\_textbox1.Remove(fnd);

break;

}

}

foreach (TextBox fnd in save\_textbox2)

{

if (fnd.Name == name)

{

save\_textbox2.Remove(fnd);

break;

}

}

foreach (TextBox fnd in save\_textbox3)

{

if (fnd.Name == name)

{

save\_textbox3.Remove(fnd);

break;

}

}

foreach (TextBox fnd in save\_textbox4)

{

if (fnd.Name == name)

{

save\_textbox4.Remove(fnd);

break;

}

}

}

else

MessageBox.Show("Элемент не удален");

}

private void save\_name(object sender, EventArgs e)

{

name = ActiveControl.Name;

if (draw == null)

{

draw = ActiveControl;

next = draw;

}

else

{

draw = next;

next = ActiveControl;

}

}

/\*private void button14\_Click(object sender, EventArgs e)

{

if (this.Width < Screen.PrimaryScreen.WorkingArea.Width)

{

this.Location = new Point(0, this.Location.Y);

this.Width = Screen.PrimaryScreen.WorkingArea.Width;

button14.Text = "→←";

}

else

{

this.Width = 1175;

this.CenterToScreen();

button14.Text = "←→";

}

}\*/

private void Form1\_Resize(object sender, EventArgs e)

{

button11.Location = new Point(this.Width - button11.Width, button11.Location.Y);

button8.Location = new Point(this.Width - 12 - button8.Width, button8.Location.Y);

}

private void button15\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Вы уверены, что хотите удалить все элементы структуры?", "Очистить форму?", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

SuspendLayout();

for (int i = Controls.Count - 1; i >= 0; i--)

{

Control Ctrl1 = Controls[i];

if (Ctrl1.Name.Contains("Элемент управления"))

{

Controls.RemoveAt(i);

Ctrl1.Dispose();

}

}

Refresh();

ResumeLayout();

save\_textbox1.RemoveRange(0, save\_textbox1.Count);

save\_textbox2.RemoveRange(0, save\_textbox2.Count);

save\_textbox3.RemoveRange(0, save\_textbox3.Count);

save\_textbox4.RemoveRange(0, save\_textbox4.Count);

for (int i = 0; i < linee.Count(); i++)

linee[i] = null;

}

}

//печать

Bitmap memoryImage;

private void printDocument1\_PrintPage(System.Object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(memoryImage, 0, 0);

}

private void CaptureScreen()

{

Graphics myGraphics = this.CreateGraphics();

memoryImage = new Bitmap(this.Width, this.Height, myGraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, this.Size);

}

private void button12\_Click(object sender, EventArgs e)

{

//печать

CaptureScreen();

printDocument1.DefaultPageSettings.Landscape = true;

printDocument1.Print();

}

private void hide()

{

foreach (Control pr in this.Controls)

{

if (pr.GetType() == typeof(Button))

pr.Visible = false;

if (pr.GetType() == typeof(CheckBox))

pr.Visible = false;

}

}

private void show()

{

foreach (Control pr in this.Controls)

{

if (pr.GetType() == typeof(Button))

pr.Visible = true;

if (pr.GetType() == typeof(CheckBox))

pr.Visible = true;

}

}

private void button13\_Click(object sender, EventArgs e)

{

//сохранение картинки

hide();

Graphics myGraphics = this.CreateGraphics();

memoryImage = new Bitmap(this.Width, this.Height, myGraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, this.Size);

memoryImage.Save(@"..\..\Diagrams\diagram.jpg", ImageFormat.Bmp);

show();

}

}

public class str

{

public System.Windows.Forms.TextBox PrevBox = null;

public System.Windows.Forms.TextBox ActiveBox = null;

}

}