

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухова Татьяна Александровна

Должность: Директор Пятигорского института (филиал) Северо-Кавказского

федерального университета

Дата подписания: 13.06.2024 16:10:33

Уникальный программный ключ:

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

ПМ.06 СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

МДК.06.03 УСТРОЙСТВО И ФУНКЦИОНИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Специальности СПО

09.02.07 Информационные системы и программирование

Пятигорск 2024

Методические указания для практических занятий по дисциплине МДК.06.03 Устройство и функционирование информационной системы составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.07 Информационные системы и программирование.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В результате освоения учебной дисциплины обучающийся должен: **знать**: регламенты и нормы по обновлению и техническому сопровождению обслуживаемой информационной системы; политику безопасности в современных информационных системах; достижения мировой и отечественной информатики в области интеллектуализации информационных систем; принципы работы экспертных систем.

уметь: осуществлять настройку информационной системы для пользователя согласно технической документации; применять основные правила и документы системы сертификации Российской Федерации; применять основные технологии экспертных систем; разрабатывать обучающие материалы для пользователей по эксплуатации информационных систем.

иметь практический опыт в: установке, настройка и сопровождение информационной системы; выполнении регламентов по обновлению, техническому сопровождению и восстановлению данных информационной системы.

В результате освоения учебной дисциплины студент должен овладевать:

Общими компетенциями:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

Профессиональными компетенциями:

ПК 6.1. Разрабатывать техническое задание на сопровождение информационной системы.

ПК 6.2. Выполнять исправление ошибок в программном коде информационной системы.

ПК 6.3. Разрабатывать обучающую документацию для пользователей информационной системы.

ПК 6.4. Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания.

ПК 6.5. Осуществлять техническое сопровождение, обновление и восстановление данных информационной системы в соответствии с техническим заданием.

ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

По дисциплине «Устройство и функционирование информационной системы» практические работы содержат задачи и теоретические вопросы. Варианты для каждого обучающегося – индивидуальные.

Задачи и ответы на вопросы, выполненные не по своему варианту, не засчитываются.

Практические занятия выполняются в отдельной тетради. Условия задачи и формулировки вопросов переписываются полностью. Формулы, расчеты, ответы на вопросы пишутся ручкой, а чертежи, схемы и рисунки выполняются карандашом, на графиках и диаграммах указывается масштаб. Вначале задача решается в общем виде, затем делаются расчёты по условию задания. Решение задач обязательно ведется в Международной системе единиц (СИ).

При выполнении практического занятия необходимо следовать методическим указаниям: повторить краткое содержание теории, запомнить основные формулы и законы, проанализировать пример выполнения аналогичного задания, затем приступить непосредственно к решению задачи. К зачету допускаются студенты, получившие положительные оценки по всем практическим работам.

Правила выполнения практических работ.

1. Студент должен прийти на практическое занятие подготовленным к выполнению практической работы.
2. Каждый студент после проведения работы должен представить отчет о проделанной работе с анализом полученных результатов и выводом по работе.
3. Таблицы и рисунки следует выполнять с помощью чертежных инструментов (линейки, циркуля, и т.д.) карандашом с соблюдением ЕСКД.
4. Расчет следует проводить с точностью до двух значащих цифр.
5. Исправления проводить на обратной стороне листа. При мелких исправлениях неправильное слово (буква, число и т.п.) аккуратно зачеркивается и над ним пишут правильное пропущенное слово (букву, число и т.п.).
6. Вспомогательные расчеты можно выполнять на отдельных листах, а при необходимости на листах отчета.
7. Если студент не выполнит практическую работу или часть работы, то он выполнит ее во внеурочное время, согласованное с преподавателем.
8. Оценку по практической работе студент получает с учетом срока выполнения работы, если;
 - расчеты выполнены правильно и в полном объеме;
 - сделан анализ проделанной работы и вывод по результатам работы;
 - студент может пояснить выполнение любого этапа работы;
 - отчет выполнен в соответствии с требованиями к выполнению работы.

Практическая работа №1.

Классификация информационных систем

Цель: Изучить основные термины и понятия, характеристики, классификацию информационных систем.

Понятие информационной системы

Под **системой** понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов. Системы значительно отличаются между собой как по составу, так и по главным целям.

Система	Элементы системы	Главная цель системы
Фирма	Люди, оборудование, материалы, здания и др.	Производство товаров

Компьютер	Электронные и электромеханические элементы, Обработка данных линии связи и др.
Телекоммуникационная система	Компьютеры, модемы, кабели, сетевое программное обеспечение и Передача информации др.
Информационная система	Компьютеры, компьютерные сети, Производство люди, информационное и профессиональной программное обеспечение информации

В информатике понятие "система" широко распространено и имеет множество смысловых значений. Чаще всего оно используется применительно к набору технических средств и программ. Системой может называться аппаратная часть компьютера. Системой может также считаться множество программ для решения конкретных прикладных задач, дополненных процедурами ведения документации и управления расчетами.

Добавление к понятию "система" слова "информационная" отражает цель ее создания и функционирования. Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые продукты.

Информационная система - взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации персонального компьютера. В крупных организациях наряду с персональным компьютером в состав технической базы информационной системы может входить мэйнфрейм или суперЭВМ. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Необходимо понимать разницу между компьютерами и информационными системами. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем. Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

Процессы в информационной системе

Процессы, обеспечивающие работу информационной системы любого назначения, условно можно представить в виде схемы, состоящей из блоков:

- ввод информации из внешних или внутренних источников;
- обработка входной информации и представление ее в удобном виде;
- вывод информации для представления потребителям или передачи в другую систему;
- обратная связь - это информация, переработанная людьми данной организации для коррекции входной информации.

Информационная система определяется следующими свойствами:

- любая информационная система может быть подвергнута анализу, построена и управляема на основе общих принципов построения систем;
- информационная система является динамичной и развивающейся;
- при построении информационной системы необходимо

использовать системный подход;

- выходной продукцией информационной системы является информация, на основе которой принимаются решения;
- информационную систему следует воспринимать как человекокомпьютерную систему обработки информации.

В настоящее время сложилось мнение об информационной системе как о системе, реализованной с помощью компьютерной техники. Хотя в общем случае информационную систему можно понимать и в некомпьютерном варианте.

Чтобы разобраться в работе информационной системы, необходимо понять суть проблем, которые она решает, а также организационные процессы, в которые она включена. Так, например, при определении возможности компьютерной информационной системы для поддержки принятия решений следует учитывать:

- структурированность решаемых управленческих задач;
- уровень иерархии управления фирмой, на котором решение должно быть принято;
- принадлежность решаемой задачи к той или иной функциональной сфере бизнеса;
- вид используемой информационной технологии.



Технология работы в компьютерной информационной системе доступна для понимания специалистом некомпьютерной области и может быть успешно использована для контроля процессов профессиональной деятельности и управления ими.

Что можно ожидать от внедрения информационных систем Внедрение информационных систем может способствовать:

- получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов и интеллектуальных систем и т.д.;
- освобождению работников от рутинной работы за счет ее автоматизации;
- обеспечению достоверности информации;
- замене бумажных носителей данных на магнитные диски или ленты, что приводит к более рациональной организации переработки информации на компьютере и снижению объемов документов на бумаге;
- совершенствованию структуры потоков информации и системы документооборота в фирме;
- уменьшению затрат на производство продуктов и услуг;
- предоставлению потребителям уникальных услуг;
- отысканию новых рыночных ниш;
- привязке к фирме покупателей и поставщиков за счет предоставления им разных скидок и услуг.

Практическая работа №2.

Роль структуры управления в информационной системе

Цель: Изучить основные термины и понятия, характеристики, классификацию информационных систем.

Роль структуры управления в информационной системе

Создание и использование информационной системы для любой организации нацелены на решение следующих задач.

1. Структура информационной системы, ее функциональное назначение должны соответствовать целям, стоящим перед организацией. Например, в коммерческой фирме - эффективный бизнес; в государственном предприятии - решение социальных и экономических задач.

2. Информационная система должна контролироваться людьми, ими пониматься и использоваться в соответствии с основными социальными и этическими принципами.

3. Производство достоверной, надежной, своевременной и систематизированной информации.

Построение информационной системы можно сравнить с постройкой дома. Кирпичи, гвозди, цемент и прочие материалы, сложенные вместе, не дают дома. Нужны проект, землеустройство, строительство и др., чтобы появился дом.

Аналогично для создания и использования информационной системы необходимо сначала понять структуру, функции и политику организации, цели управления и принимаемых решений, возможности компьютерной технологии. Информационная система является частью организации, а ключевые элементы любой организации - структура и органы управления, стандартные процедуры, персонал, субкультура.

Построение информационной системы должно начинаться с анализа структуры управления организацией.

Аналогично для создания и использования информационной системы необходимо сначала понять структуру, функции и политику организации, цели управления и принимаемых решений, возможности компьютерной технологии. Информационная система является частью организации, а ключевые элементы любой организации - структура и органы управления, стандартные процедуры, персонал, субкультура.

Построение информационной системы должно начинаться с анализа структуры управления организацией.

Структура управления организацией

Координация работы всех подразделений организации осуществляется через органы управления разного уровня. Под **управлением** понимают обеспечение поставленной цели при условии реализации следующих функций: организационной, плановой, учетной, анализа, контрольной, стимулирования.

Рассмотрим содержание **управленческих функций**:

Организационная функция заключается в разработке организационной структуры и комплекса нормативных документов: штатного расписания фирмы, отдела, лаборатории, группы и т.п. с указанием подчиненности, ответственности, сферы компетенции, прав, обязанностей и т.п. Чаще всего это излагается в положении по отделу, лаборатории или должностных инструкциях.

Планирование (плановая функция) состоит в разработке и реализации планов по выполнению поставленных задач. Например, бизнесплан для всей фирмы, план производства, план маркетинговых исследований, финансовый план, план проведения научно-исследовательской работы и т.д. на различные сроки (год, квартал, месяц, день).

Учетная функция заключается в разработке или использовании уже готовых форм и методов учета показателей деятельности фирмы: бухгалтерский учет, финансовый учет, управленческий учет и т.п. В общем случае учет можно определить как получение, регистрацию, накопление, обработку и предоставление информации о реальных хозяйственных процессах.

Анализ или аналитическая функция связывается с изучением итогов выполнения планов и заказов, определением влияющих факторов, выявлением резервов, изучением тенденций развития и т.д. Выполняется анализ разными специалистами в зависимости от сложности и уровня анализируемого объекта или процесса. Анализ результатов хозяйственной деятельности фирмы за год и более проводят специалисты, а на уровне цеха, отдела $\approx$ менеджеры этого уровня (начальник или его заместитель) совместно со специалистом-экономистом.

Контрольная функция чаще всего осуществляется менеджером: контроль за выполнением планов, расходованием материальных ресурсов, использованием финансовых средств и т.п.

Стимулирование или мотивационная функция предполагает разработку и применение различных методов стимулирования труда подчиненных работников:

- финансовые стимулы - зарплата, премия, акции, повышение в должности и т.п.;
- психологические стимулы - благодарности, грамоты, звания, степени, доски почета и т.п.

В последние годы в сфере управления все активнее стали применяться понятие "принятие решения" и связанные с этим понятием системы, методы, средства поддержки принятия решений.

Принятие решения - акт целенаправленного воздействия на объект управления, основанный на анализе ситуации, определении цели, разработке программы достижения этой цели.

Структура управления любой организации традиционно делится на три уровня: операционный, функциональный и стратегический.

Уровни управления (вид управленческой деятельности) определяются сложностью решаемых задач. Чем сложнее задача, тем более высокий уровень управления требуется для ее решения. При этом следует понимать, что более простых задач, требующих немедленного (оперативного) решения, возникает значительно большее количество, а значит, и уровень управления для них нужен другой - более низкий, где принимаются решения оперативно. При управлении необходимо также учитывать динамику реализации принимаемых решений, что позволяет рассматривать управление под углом временного фактора.

На рис ниже отображены три уровня управления, которые соотнесены с такими факторами, как степень возрастания власти, ответственности, сложности решаемых задач, а также динамика принятия решений по реализации задач.



Операционный (нижний) уровень управления обеспечивает решение многократно повторяющихся задач и операций и быстрое реагирование на изменения входной текущей информации. На этом уровне достаточно велики как объем выполняемых операций, так и динамика принятия управленческих решений. Этот уровень управления часто называют оперативным из-за необходимости быстрого реагирования на изменение ситуации. На уровне оперативного (операционного) управления большой объем занимают учетные задачи.

Функциональный (тактический) уровень управления обеспечивает решение задач, требующих предварительного анализа информации, подготовленной на первом уровне. На этом уровне большое значение приобретает такая функция управления, как анализ. Объем решаемых задач уменьшается, но возрастает их сложность. При этом не всегда удастся выработать нужное решение оперативно, требуется дополнительное время на анализ, осмысление, сбор недостающих сведений и т.п. Управление связано с некоторой задержкой от момента поступления информации до принятия решений и их реализации, а также от момента реализации решений до получения реакции на них.

Стратегический уровень обеспечивает выработку управленческих решений, направленных на достижение долгосрочных стратегических целей организации. Поскольку результаты принимаемых решений проявляются спустя длительное время, особое значение на этом уровне имеет такая функция управления, как стратегическое планирование. Прочие функции управления на этом уровне в настоящее время разработаны недостаточно полно. Часто стратегический уровень управления называют стратегическим или долгосрочным планированием. Правомочность принятого на этом уровне решения может быть подтверждена спустя достаточно длительное время. Могут пройти месяцы или годы. Ответственность за принятие управленческих решений чрезвычайно велика и определяется не только результатами анализа с использованием математического и специального аппарата, но и профессиональной интуицией менеджеров.

Практическая работа №3.

Структура и классификация информационных систем.

Цель: Изучить основные термины и понятия, характеристики, классификацию информационных систем.

Структура и классификация информационных систем

Структура информационной системы

Типы обеспечивающих подсистем

Структуру информационной системы составляет совокупность отдельных ее частей, называемых подсистемами.

Подсистема - это часть системы, выделенная по какому-либо признаку.

Общую структуру информационной системы можно рассматривать как совокупность подсистем независимо от сферы применения. В этом случае говорят о **структурном признаке классификации**, а подсистемы называют обеспечивающими. Таким образом, структура любой информационной системы может быть представлена совокупностью обеспечивающих подсистем



Среди обеспечивающих подсистем обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение

Назначение подсистемы информационного обеспечения состоит в современном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение - совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

Унифицированные системы документации создаются на государственном, республиканском, отраслевом и региональном уровнях.

Главная цель - это обеспечение сопоставимости показателей различных сфер общественного производства. Разработаны стандарты, где устанавливаются требования:

- к унифицированным системам документации;
- к унифицированным формам документов различных уровней управления;
- к составу и структуре реквизитов и показателей;
- к порядку внедрения, ведения и регистрации унифицированных форм документов.

Однако, несмотря на существование унифицированной системы документации, при обследовании большинства организаций постоянно выявляется целый комплекс типичных недостатков:

- чрезвычайно большой объем документов для ручной обработки; □
одни и те же показатели часто дублируются в разных документах;

- работа с большим количеством документов отвлекает

специалистов от решения непосредственных задач;

- имеются показатели, которые создаются, но не используются, и др.

Поэтому устранение указанных недостатков является одной из задач, стоящих при создании информационного обеспечения.

Схемы информационных потоков отражают маршруты движения информации и ее объемы, места возникновения первичной информации и использования результатной информации. За счет анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Построение схем информационных потоков, позволяющих выявить объемы информации и провести ее детальный анализ, обеспечивает:

- исключение дублирующей и неиспользуемой информации; □
классификацию и рациональное представление информации.

Методология построения баз данных базируется на теоретических основах их проектирования. Для понимания концепции методологии приведем основные ее идеи в виде двух последовательно реализуемых на практике этапов:

1-й этап - обследование всех функциональных подразделений фирмы с целью:

- понять специфику и структуру ее деятельности;
- построить схему информационных потоков;
- проанализировать существующую систему документооборота;
- определить информационные объекты и соответствующий состав реквизитов (параметров, характеристик), описывающих их свойства и назначение.

2-й этап - построение концептуальной информационно-логической модели данных для обследованной на 1-м этапе сферы деятельности. В этой модели должны быть установлены и оптимизированы все связи между объектами и их реквизитами. Информационно-логическая модель является фундаментом, на котором будет создана база данных.

- ясное понимание целей, задач, функций всей системы

управления организацией;

- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;

- совершенствование системы документооборота;
- наличие и использование системы классификации и

кодирования;

- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;

- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

Техническое обеспечение

Техническое обеспечение - комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации; эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Перспективным подходом следует считать, по-видимому, частично децентрализованный подход - организацию технического обеспечения на базе распределенных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение

Математическое и программное обеспечение - совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

К средствам **математического обеспечения** относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав **программного обеспечения** входят общесистемные и специальные программные продукты, а также техническая документация.

К **общесистемному программному обеспечению** относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономикоматематическую модель задачи, контрольные примеры.

Организационное обеспечение

Организационное обеспечение — это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации ИС.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на 1-м этапе построения БД. **Правовое обеспечение**

Правовое обеспечение - совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности.

В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования информационной системы включает:

- статус информационной системы;
- права, обязанности и ответственность персонала; □
порядок создания и использования информации и др.

Классификация информационных систем по признаку структурированности задач

Понятие структурированности задач

При создании или при классификации информационных систем неизбежно возникают проблемы, связанные с формальным - математическим и алгоритмическим описанием решаемых задач. От степени формализации во многом зависят эффективность работы всей системы, а также уровень автоматизации, определяемый степенью участия человека при принятии решения на основе получаемой информации.

Чем точнее математическое описание задачи, тем выше возможности компьютерной обработки данных и тем меньше степень участия человека в процессе ее решения. Это и определяет степень автоматизации задачи.

Различают три типа задач, для которых создаются информационные системы: структурированные (формализуемые), неструктурированные (не формализуемые) и частично структурированные.

Структурированная (формализуемая) задача - задача, где известны все ее элементы и взаимосвязи между ними.

Неструктурированная (не формализуемая) задача - задача, в которой невозможно выделить элементы и установить между ними связи.

В структурированной задаче удастся выразить ее содержание в форме математической модели, имеющей точный алгоритм решения. Подобные задачи обычно приходится решать многократно, и они носят рутинный характер. Целью использования информационной системы для решения структурированных задач является полная автоматизация их решения, т.е. сведение роли человека к нулю.

Типы информационных систем, используемые для решения частично структурированных задач

Информационные системы, используемые для решения частично структурированных задач, подразделяются на два вида создающие управленческие отчеты и ориентированные главным образом на обработку данных (поиск, сортировку, агрегирование, фильтрацию). Используя сведения, содержащиеся в этих отчетах, управляющий принимает решение;



Информационные системы, **создающие управленческие отчеты**, обеспечивают информационную поддержку пользователя, т.е. предоставляют доступ к информации в базе данных и ее частичную обработку. Процедуры манипулирования данными в информационной системе должны обеспечивать следующие возможности:

- составление комбинаций данных, получаемых из различных источников;
- быстрое добавление или исключение того или иного источника данных и автоматическое переключение источников при поиске данных;
- управление данными с использованием возможностей систем управления базами данных;
- логическую независимость данных этого типа от других баз данных, входящих в подсистему информационного обеспечения;
- автоматическое отслеживание потока информации для наполнения баз данных.

Информационные системы, **разрабатывающие альтернативы решений**, могут быть модельными и экспертными.

Модельные информационные системы предоставляют пользователю математические, статические, финансовые и другие модели, использование которых облегчает выработку и оценку альтернатив решения. Пользователь может получить

недостающую ему для принятия решения информацию путем установления диалога с моделью в процессе ее исследования.

Основными функциями модельной информационной системы являются:

- возможность работы в среде типовых математических моделей, включая решение основных задач моделирования типа "как сделать, чтобы?", "что будет, если?", анализ чувствительности и др.;
- достаточно быстрая и адекватная интерпретация результатов моделирования;
- оперативная подготовка и корректировка входных параметров и ограничений модели;
- возможность графического отображения динамики модели;
- возможность объяснения пользователю необходимых шагов формирования и работы модели.

Экспертные информационные системы обеспечивают выработку и оценку возможных альтернатив пользователем за счет создания экспертных систем, связанных с обработкой знаний. Экспертная поддержка принимаемых пользователем решений реализуется на двух уровнях.

Работа первого уровня экспертной поддержки исходит из концепции "типовых управленческих решений", в соответствии, с которой часто возникающие в процессе управления проблемные ситуации можно свести к некоторым однородным классам управленческих решений, т.е. к некоторому типовому набору альтернатив. Для реализации экспертной поддержки на этом уровне создается информационный фонд хранения и анализа типовых альтернатив.

Если возникшая проблемная ситуация не ассоциируется с имеющимися классами типовых альтернатив, в работу должен вступать второй уровень экспертной поддержки управленческих решений. Этот уровень генерирует альтернативы на базе имеющихся в информационном фонде данных, правил преобразования и процедур оценки синтезированных альтернатив.

Практическая работа №4.

Прочие классификации информационных систем

Цель: Изучить основные термины и понятия, характеристики, классификацию информационных систем.

Прочие классификации информационных систем

Классификация по степени автоматизации

В зависимости от степени автоматизации информационных процессов в системе управления фирмой информационные системы определяются как ручные, автоматические, автоматизированные

Ручные ИС характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС.

Автоматические ИС выполняют все операции по переработке информации без участия человека.

Автоматизированные ИС предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится

компьютеру. В современном толковании в термин "информационная система" вкладывается обязательно понятие автоматизируемой системы.

Автоматизированные ИС, учитывая их широкое использование в организации процессов управления, имеют различные модификации и могут быть классифицированы, например, по характеру использования информации и по сфере применения.

Информационно-поисковые системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Например, информационно-поисковая система в библиотеке, в железнодорожных и авиа кассах продажи билетов.

Информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Среди них можно провести классификацию по степени воздействия выработанной результатной информации на процесс принятия решений и выделить два класса: управляющие и советующие.

Управляющие ИС вырабатывают информацию, на основании которой человек принимает решение. Для этих систем характерны тип задач расчетного характера и обработка больших объемов данных. Примером могут служить система оперативного планирования выпуска продукции, система бухгалтерского учета.

Советующие ИС вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.



Классификация по характеру использования информации

Классификация по сфере применения

Информационные системы **организационного управления** предназначены для автоматизации функций управленческого персонала. Учитывая наиболее широкое применение и разнообразие этого класса систем, часто любые информационные системы понимают именно в данном толковании. К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми фирмами и др. Основными функциями подобных систем являются: оперативный контроль и регулирование, оперативный учет и анализ,

перспективное и оперативное планирование, бухгалтерский учет, управление сбытом и снабжением и другие экономические и организационные задачи.

ИС управления технологическими процессами (ТП) служат для автоматизации функций производственного персонала. Они широко используются при организации для поддержания технологического процесса в металлургической и машиностроительной промышленности.

ИС автоматизированного проектирования (САПР) предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются: инженерные расчеты, создание графической документации (чертежей, схем, планов), создание проектной документации, моделирование проектируемых объектов.

Интегрированные (корпоративные) ИС используются для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта продукции. Создание таких систем весьма затруднительно, поскольку требует системного подхода с позиций главной цели, например получения прибыли, завоевания рынка сбыта и т.д. Такой подход может привести к существенным изменениям в самой структуре фирмы, на что может решиться не каждый управляющий.

Документальные информационные системы

В отличие от фактографических информационных систем, единичным элементом данных в документальных [информационных системах] является неструктурированный на более мелкие элементы документ. В качестве неструктурированных документов в подавляющем большинстве случаев выступают, прежде всего, текстовые документы, представленные в виде текстовых файлов, хотя к классу неструктурированных документированных данных могут также относиться звуковые и графические файлы.

Основной задачей документальных информационных систем является накопление и предоставление пользователю документов, содержание, тематика, реквизиты и т. п. которых адекватны его информационным потребностям. Поэтому можно дать следующее **определение документальной информационной системы** — единое хранилище документов с инструментарием поиска и отбора необходимых документов. Поисковый характер документальных информационных систем исторически определил еще одно их название — информационно-поисковые системы (ИПС), хотя этот термин не совсем полно отражает специфику документальных ИС.

Соответствие найденных документов информационным потребностям пользователя называется **пертинентностью**.

В силу теоретических и практических сложностей с формализацией смыслового содержания документов пертинентность относится скорее к качественным понятиям, хотя, как будет рассмотрено ниже, может выражаться определенными количественными показателями.

В зависимости от особенностей реализации хранилища документов и механизмов поиска документальные ИПС можно разделить на две группы: □ системы на основе индексирования; □ семантически-навигационные системы.

В **семантически-навигационных системах** документы, помещаемые в хранилище (в базу) документов, оснащаются специальными навигационными конструкциями, соответствующими смысловым связям (отсылкам) между различными документами или отдельными фрагментами одного документа. Такие конструкции реализуют некоторую семантическую* (смысловую) сеть в базе документов. Способ и механизм выражения информационных потребностей в подобных системах заключаются

в явной навигации пользователя по смысловым отсылкам между документами. В настоящее время такой подход реализуется в гипертекстовых ИПС.

В системах на основе индексирования исходные документы помещаются в базу без какого-либо дополнительного преобразования, но при этом смысловое содержание каждого документа отображается в некоторое поисковое пространство. Процесс отображения документа в поисковое пространство называется индексированием и заключается в присвоении каждому документу некоторого индекса-координаты в поисковом пространстве. Формализованное представление (описание) индекса документа называется поисковым образом документа (ПОД). Пользователь выражает свои информационные потребности средствами и языком поискового пространства, формируя поисковый образ запроса (ПОЗ) к базе документов. Система на основе определенных критериев и способов ищет документы, поисковые образы которых соответствуют или близки поисковым образам запроса пользователя, и выдает соответствующие документы. Соответствие найденных документов запросу пользователя называется релевантностью. Схематично общий принцип устройства и функционирования документальных ИПС на основе индексирования иллюстрируется на рис.

Рис. Общий принцип устройства и функционирования документальных ИПС на основе индексирования

Особенностью документальных ИПС является также то, что в их функции, как правило, включаются и задачи информационного оповещения пользователей по всем новым поступающим в систему документам, соответствующим заранее определенным информационным потребностям пользователя.

Принцип решения задач информационного оповещения в документальных ИПС на основе индексирования аналогичен принципу решения задач поиска документов по запросам и основан на отображении в поисковое пространство информационных потребностей пользователя в виде так называемых поисковых профилей пользователей (ППП). Информационно-поисковая система по мере поступления и индексирования новых документов сравнивает их образы с поисковыми профилями пользователей и принимает решение о соответствующем оповещении.

Поисковое пространство, отображающее поисковые образы документов и реализующее механизмы информационного поиска документов так же, как и в СУБД фактографических систем, строится на основе языков документальных баз данных, называемых информационнопоисковыми языками (ИПЯ).

Информационно-поисковый язык представляет собой некоторую формализованную семантическую систему, предназначенную для выражения содержания документа и запросов по поиску необходимых документов. По аналогии с языками баз данных фактографических систем ИПЯ можно разделить на структурную и манипуляционную составляющие.

Структурная составляющая ИПЯ (поискового пространства) документальных ИПС на основе индексирования реализуется индексными указателями в форме информационно-поисковых каталогов, тезаурусов и генеральных указателей.

Информационно-поисковые каталоги являются традиционными технологиями организации информационного поиска в документальных фондах библиотек, архивов и представляют собой классификационную систему знаний по определенной предметной области. Смысловое содержание документа в информационно-поисковых каталогах отображается тем или иным классом каталога, а индексирование документов заключается в присвоении каждому документу специального кода (индекса) соответствующего по содержанию класса (классов) каталога и создания на этой основе специального индексного указателя.

Тезаурус представляет собой специальным образом организованную совокупность основных лексических единиц (понятий) предметной области (словарь терминов) и описание парадигматических отношений между ними. Парадигматические отношения выражаются семантическими отношениями между элементами словаря, не зависящими от любого контекста. Независимость от контекста означает обобщенность (абстрагированность) смысловых отношений, например отношения «род-вид», «предмет-целое», «субъект-объект-средство-место-время действия». Так же, как и в информационно-поисковых каталогах, в системах на основе тезаурусов в информационно-поисковом пространстве отображается не весь текст документа, а только лишь выраженное средствами тезауруса смысловое содержание документа.

Генеральный указатель (конкорданс) (глобальный словарь-индекс) в общем виде представляет собой перечисление всех слов (словоформ), имеющих в документах хранилища, с указанием (отсылками) координатного местонахождения каждого слова (№ документа — № абзаца — № предложения — № слова). Индексирование нового документа в таких системах производится через дополнение координатных отсылок тех словоформ генерального указателя, которые присутствуют в новом документе. Так как поисковое пространство в таких системах отражает полностью весь текст документа (все слова документа), а не только его смысловое содержание, то такие системы получили название полнотекстовых ИПС.

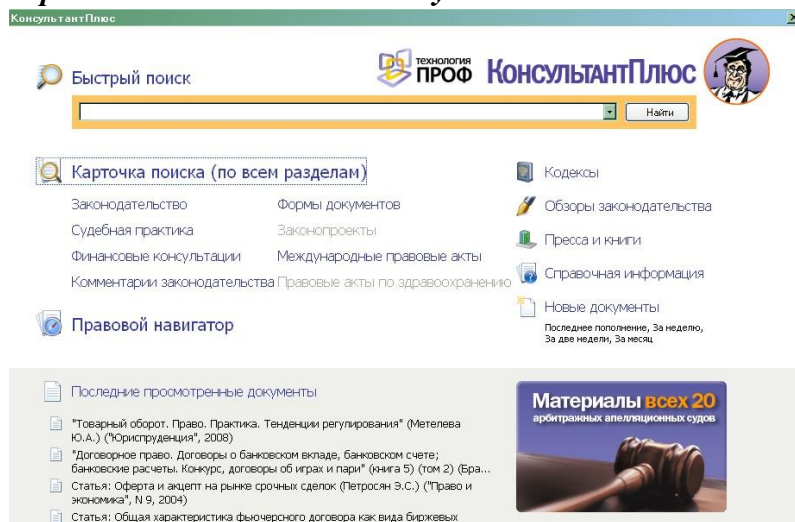
В специальной литературе такие системы иногда называют системами без лексического контроля, т. е. без учета возможной синонимичности отдельных групп словоформ, объединения отдельных групп словоформ в общие смысловые группы, семантических отношений между словоформами.

Практическая работа №5. Работа в справочно-правовой системе «Консультант-Плюс»

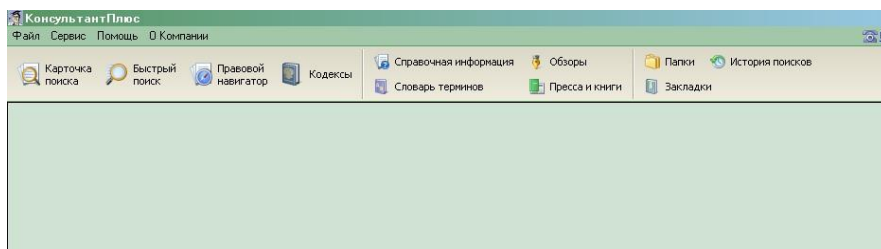
Цель: Ознакомление со структурой СПС «Консультант-Плюс», изучение основных разделов меню, получение навыков поиска правовых документов.

Описание входа в систему

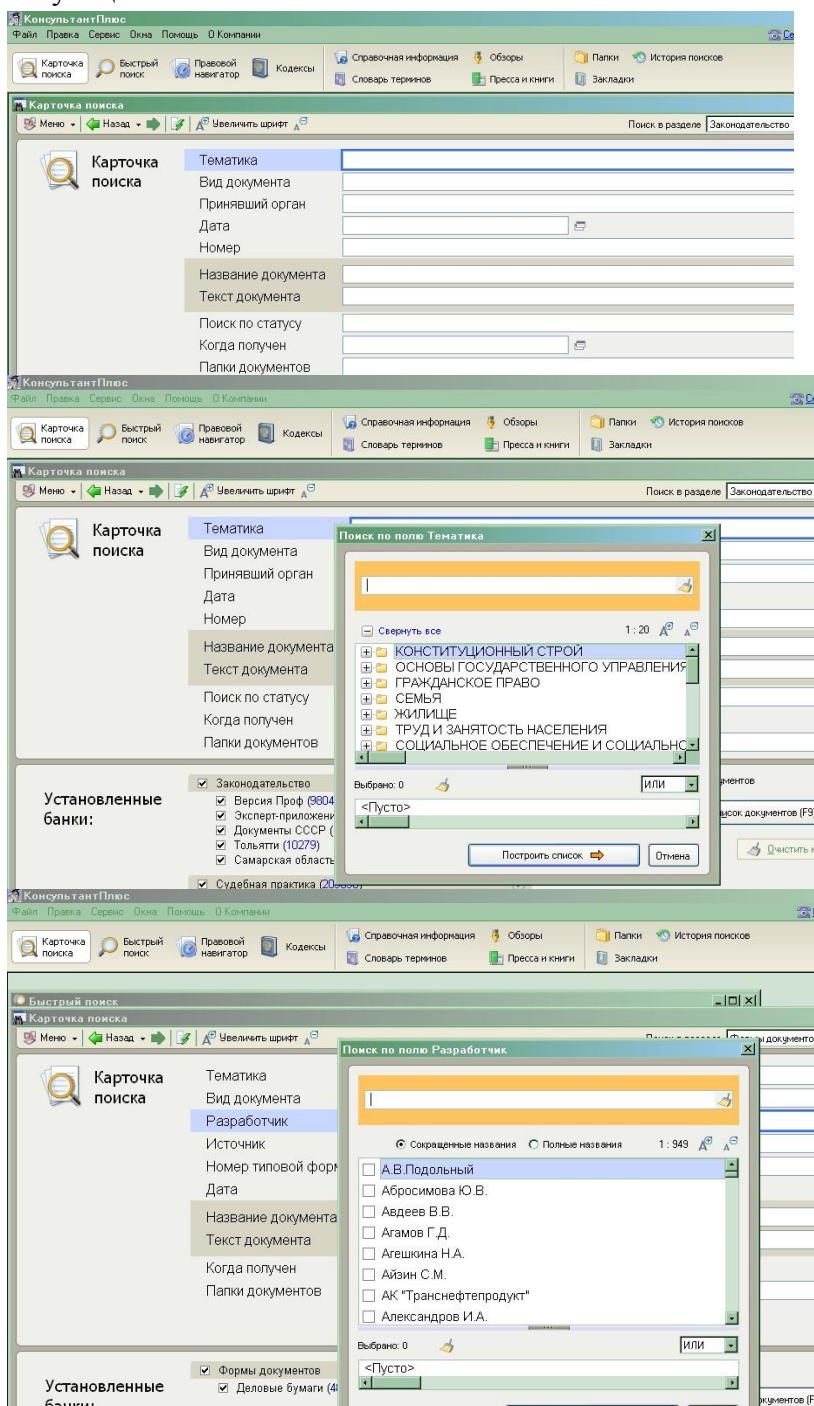
1. *Стартовое окно* в системе *Консультант-Плюс*



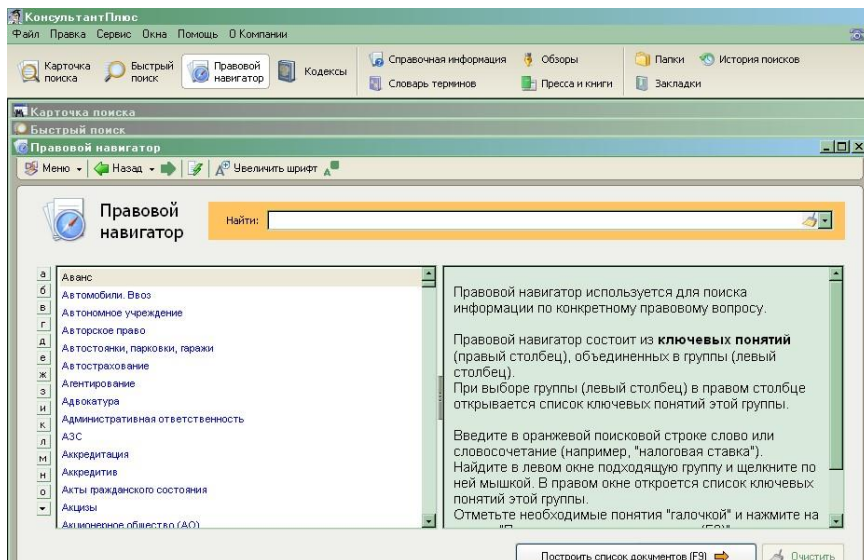
ИЛИ



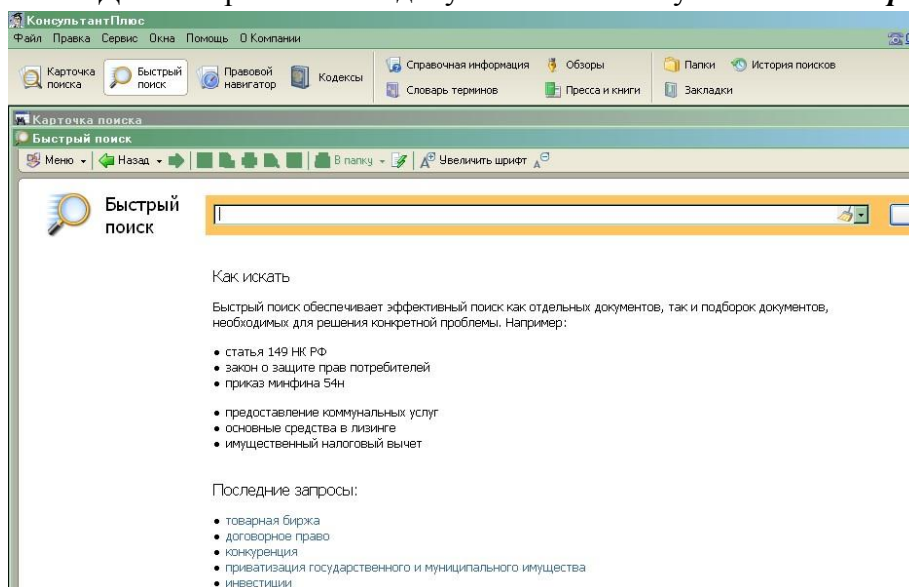
2. Для поиска документов по известным реквизитам войдите в **Карточку поиска** (по всем разделам). Для поиска документов следует ввести известные реквизиты в соответствующие поля:



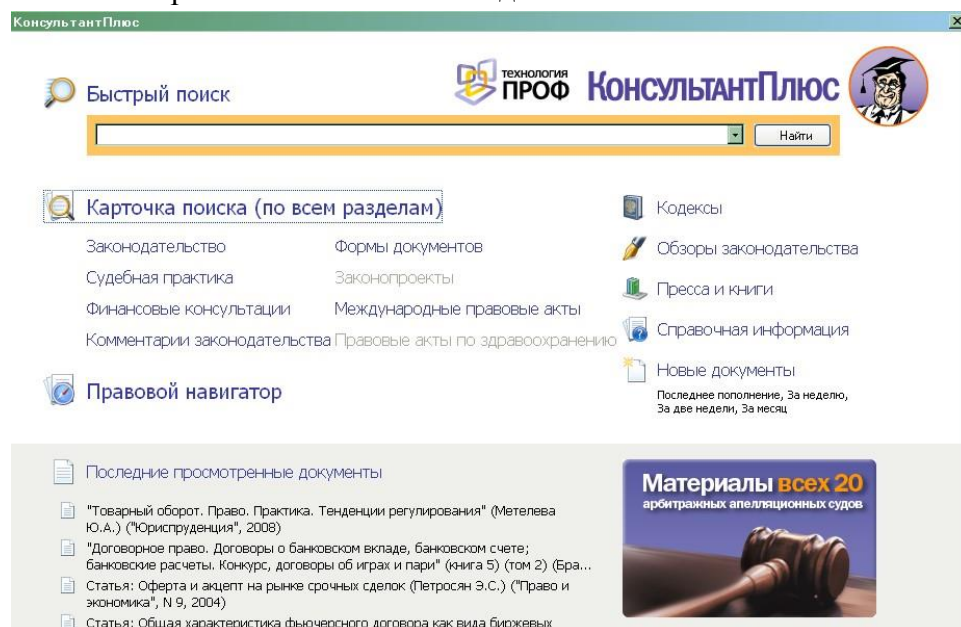
3. Для поиска информации по практическому вопросу откройте **Правовой навигатор**. Правовой навигатор позволяет найти информацию по практическому вопросу с использованием перечня возможных ситуаций.



4. Для быстрого поиска документов используют окно **Быстрого поиска**:

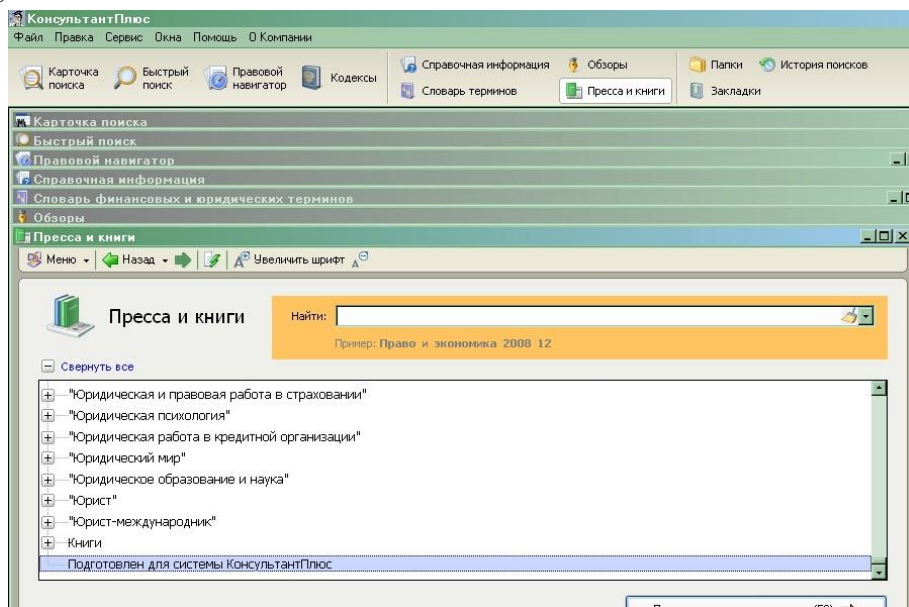


5. В Стартовом окне также находятся:



- Список последних просмотров документов;
- Переход к списку кодексов;
- Переход к новостям и обзорам законодательства;
- Поиск книг и публикаций в бухгалтерской и юридической

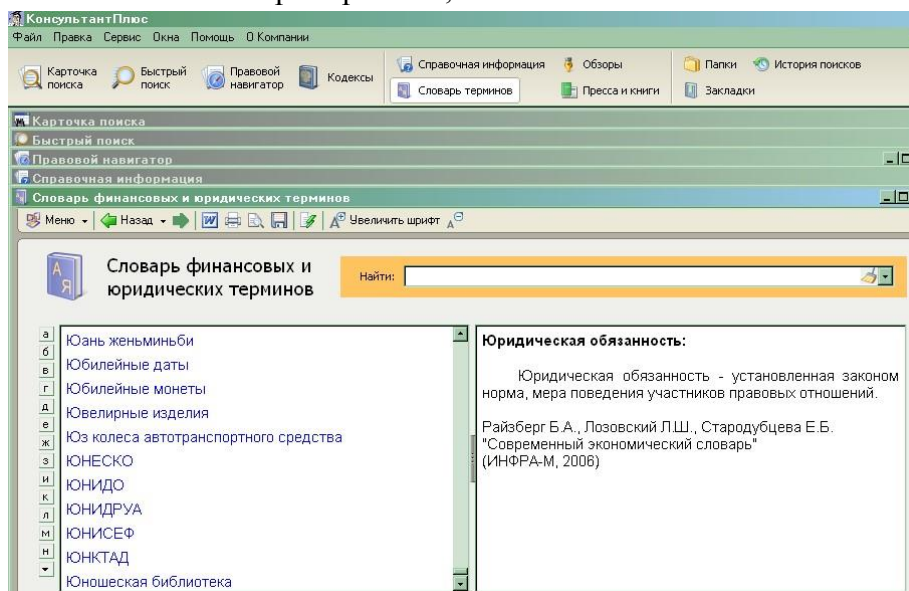
прессе;



Практическая работа №6. Представление текста в справочно-правовой системе «Консультант-плюс»

Цель: Ознакомление со структурой СПС «Консультант-Плюс», изучение основных разделов меню, получение навыков поиска правовых документов.

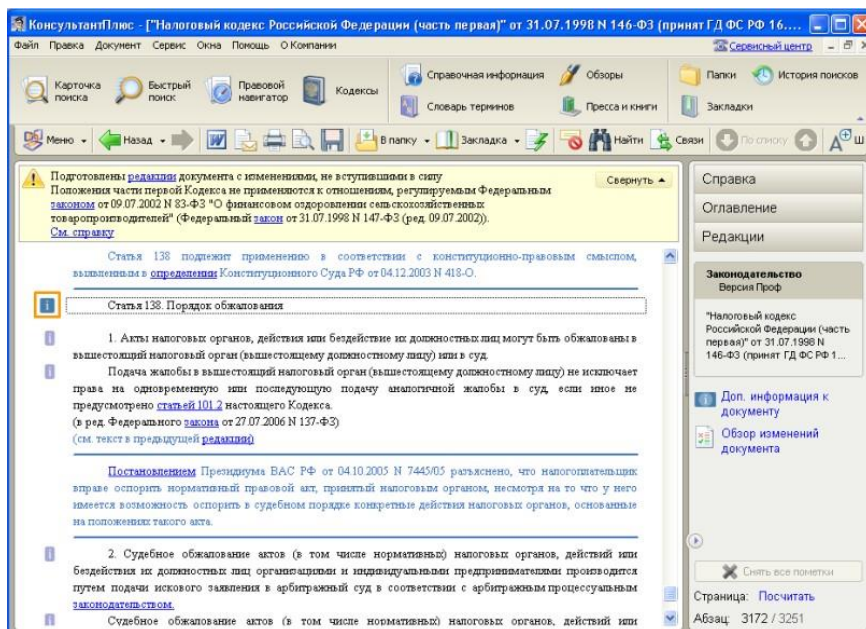
- Словарь терминов;



- Переход к справочной информации; □ Переход к документам последних наполнений; □ Переход к новостям *Консультант-Плюс*.

Представление текста в системе *Консультант-Плюс*

Информационная строка содержит информацию об особенностях применения этого документа. Располагается под названием текста документа в верхней части экрана.



Получить дополнительную информацию:

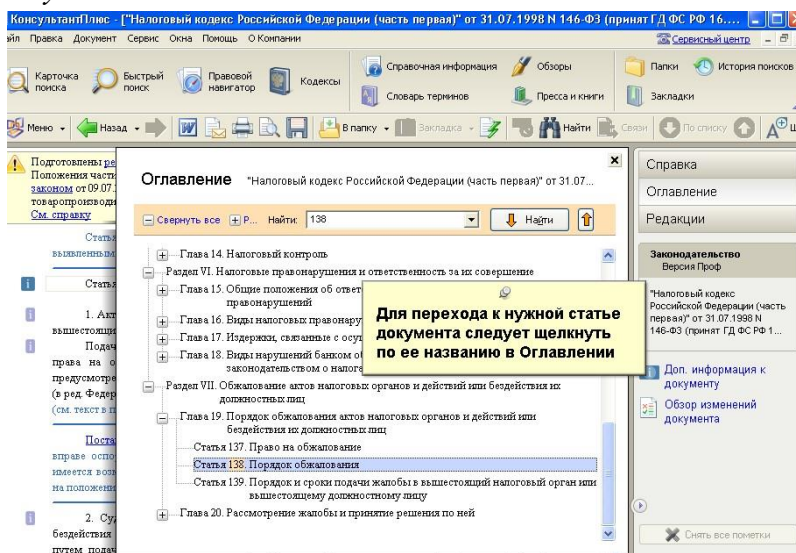
- Позволяет к документу в целом большой значок [i], расположенный в начале документа и на Правой панели.
- Ко всей статье (главе, разделу) позволяет яркий значок [i], расположенный слева от заголовка статьи (главы, раздела).
- К отдельному абзацу статьи позволяет светлый значок [i], расположенный слева от абзаца.

Примечания в тексте (отделены сплошными чертами). Для получения справки нужно щелкнуть по кнопке «Справка», откроется окно «Справка», которое содержит информацию о порядке применения, особенностях вступления в силу, источники публикации и другое.

Для перехода в оглавление документа нажмите кнопку «Оглавление», откроется окно «Оглавление», для перехода к нужной статье документа следует щелкнуть по названию в оглавлении.

Оглавление существует для документов со сложной структурой, используется для быстрого перехода в нужный раздел документа. Достаточно щелкнуть мышкой по названию пункта оглавления документа для перехода в его текст:

- Вызывается одноимённой кнопкой Правой панели окна с текстом документа.

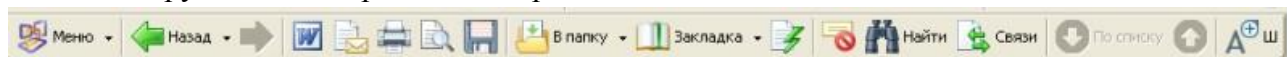


- Наличие у раздела подразделов отмечено символом слева от названия раздела, что позволяет разворачивать/сворачивать оглавление разделов документа.

На **Правой панели** также расположены:

- Список всех редакций документа;
- Обзор изменений документа;
- Подсчет страниц для печати.

Панель инструментов содержит пиктограммы:



- копирование текста в Word;
- отправка документа по почте;
- печать текста документа;
- связи данного документа с другими материалами; □ изменение размера шрифта и др.

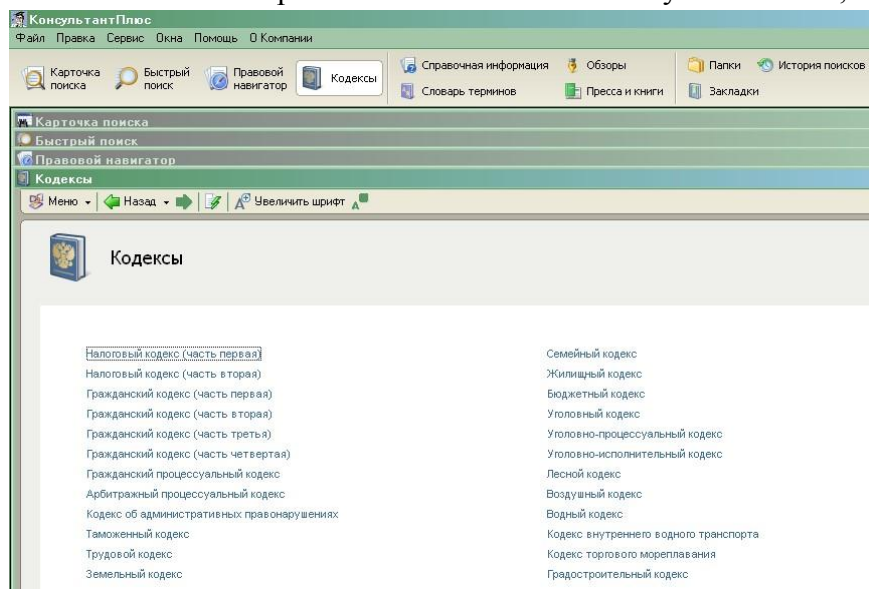
Любой документ можно распечатать, если, находясь в его тексте, нажать кнопку  пиктографического меню.

Кнопка  позволяет с помощью установленной на компьютере Почтовой программы переслать найденный в системе **Консультант-Плюс** документа в виде текстового вложения. После нажатия этой кнопки автоматически откроется окно почтовой программы, и файл с текстом документа будет прикреплён к созданному сообщению.

Нужен кодекс

Пример: найдём вторую часть налогового кодекса:

- На стартовом окне нажмите кнопку **«Кодексы»**;

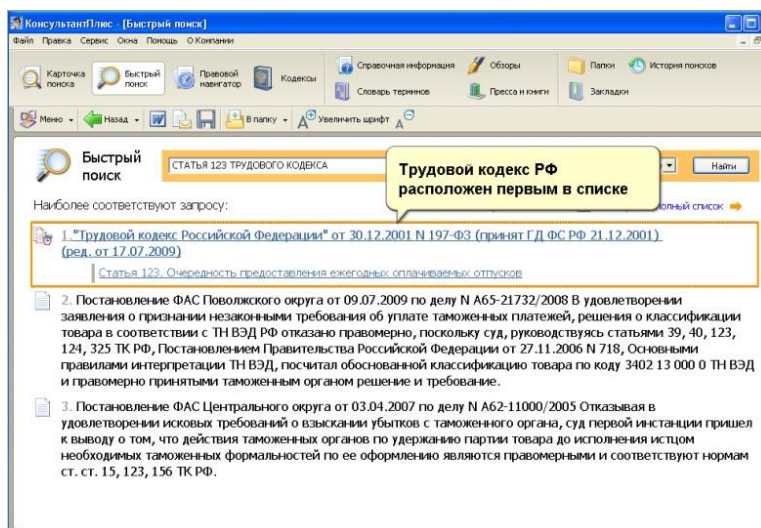


- В появившемся списке найдите **Налоговый кодекс (часть вторая)** и, щёлкнув мышью по его названию, перейдете в текст документа.

Нужна конкретная статья документа

Пример: найдём статью 123 Трудового кодекса:

- В Стартовом окне нажать кнопку **«Быстрый поиск»**;
- В строке быстрого поиска набрать: **«статья 123 Трудового кодекса»**, нажать **«Найти»**;



3. Под названием кодекса размещена ссылка на искомую статью, если щелкнуть по ссылке, документ сразу откроется на статье 123.

Как быстро войти в курс дела по конкретно ситуации:

Пример: Найдем информацию о возможности обмена приобретенного товара:

- В Стартовом окне нажмите кнопку **«Быстрый поиск»**;
- В строке Быстрого поиска зададим: **«Обмен товара»**;
- Нажать кнопку **«Найти»**;
- Откроется список документов, наиболее соответствующих запросу.

Эти документы разных типов.

- Начните изучение вопроса с Гражданского кодекса РФ: щелкните по ссылке на статью 502 непосредственно под названием кодекса изучите текст статьи, затем вернитесь в список документов, нажав кнопку **«Назад»** панели инструментов.
- Просмотрите другие документы в списке.

Полный список документов

В меню **Быстрого поиска** можно перейти к полному списку документов из всех разделов и информационных банков, касающихся рассматриваемого вопроса. Например, если нужна полная подборка консультаций или судебной практики, чтобы изучить этот вопрос во всех деталях и быть уверенным, что никакие нюансы не упущены.

Практическая работа №7.

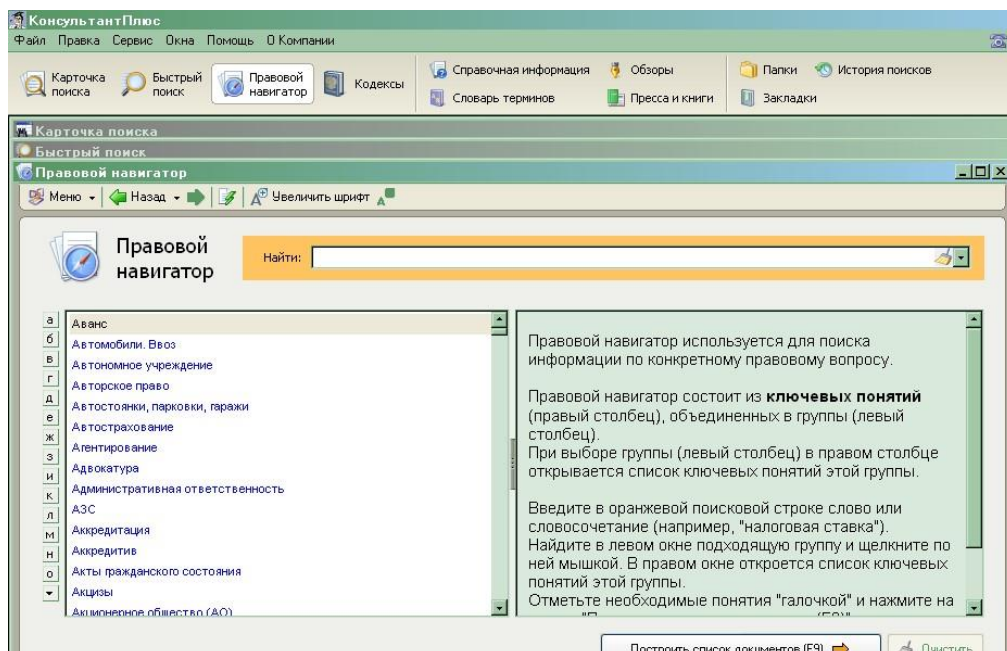
Ключевые понятия Правового навигатора, соответствующие запросу.

Цель: Ознакомление со структурой СПС «Консультант-Плюс», изучение основных разделов меню, получение навыков поиска правовых документов.

Ключевые понятия Правового навигатора, соответствующие запросу.

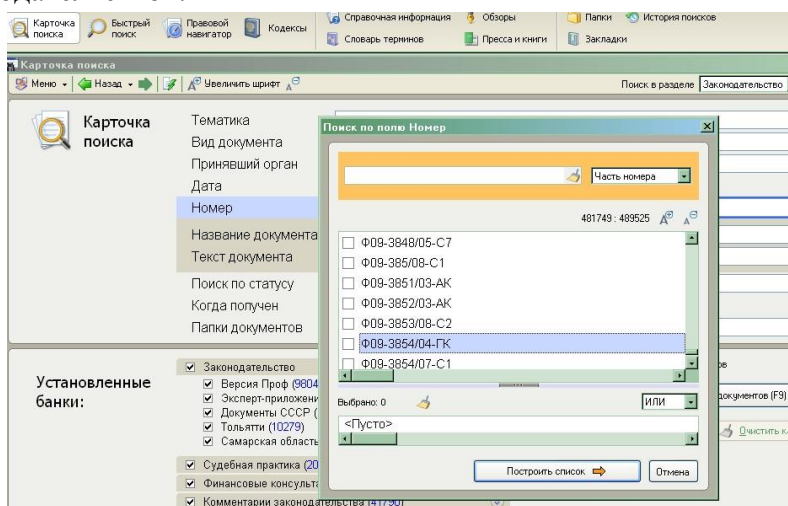
□ В списке Быстрого поиска имеется возможность ознакомиться с ключевыми понятиями Правового навигатора, соответствующими сделанному запросу.

Это удобно, так как: ○ Предложенные ключевые понятия, возможно, точнее отражают суть вашей проблемы, с их помощью можно уточнить свой запрос, получить дополнительные документы по проблеме;



○ Список этих ключевых понятий может подсказать, информация по каким смежным вопросам может быть полезна. **Если известен номер документа**

• в **Стартовом окне** системы откройте карточку поиска, кликнув по ссылке «Законодательство».



• В открывшемся окне кликните по полю «**Номер**», откроется окно «Поиск по полю Номер». В строке поиска наберём «**ПБУ**» (Положение по бухгалтерскому учёту «Учётная политика организаций (ПБУ 1/2008); (отметьте галочкой «ПБУ 1/2008»), нажмите кнопку «**Построить список (F9)**».

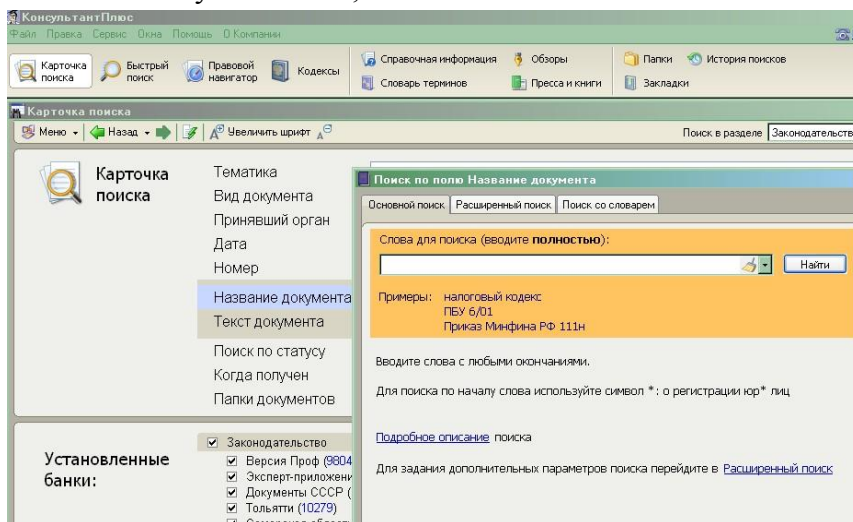
• Откроется окно со списком найденных документов по запросу (в данном случае 1), щелкните мышью по названию документа для перехода в его текст, ознакомьтесь с текстом документа.

• Искомое ПБУ 1/2008 утверждено приказом Минфина РФ от 06.10.2008 № 106н, поэтому также можно найти это ПБУ, задав в поле «**Номер**» номер утверждающего приказа (106н);

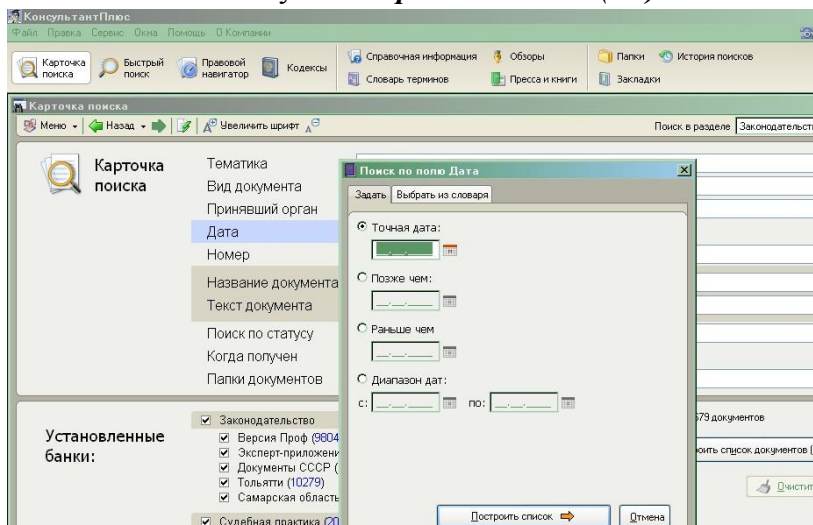
• После того как вы набрали номер в словаре поля может остаться несколько значений. В этом случае, прежде чем нажимать кнопку «**Построить список**», отметьте каждое значение галочкой. **Если примерно известны дата и название документа**

Пример: Найдём Федеральный закон о деятельности по приёму платежей физических лиц, осуществляемой платёжными агентами, принятый летом 2009 года.

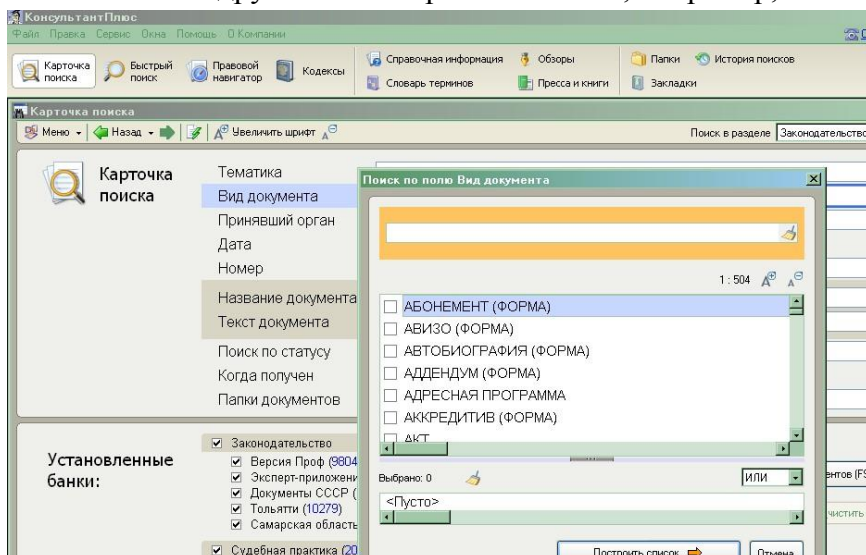
- Откройте **Карточку поиска**;
- В поле «**Название документа**» наберите: «Приём платежей физических лиц», нажмите кнопку «**Найти**»;



- Для уточнения запроса в поле «**Дата**» задайте диапазон: с 01.06.2009 по 31.08.2009. Нажмите кнопку «**Построить список (F9)**».



- Выберите в открывшемся списке нужный документ и щёлкните мышью по его названию для перехода в текст.
- Ознакомьтесь с текстом документа в открывшемся окне.
- Рекомендации по заполнению **Карточки поиска**: для уточнения запроса можно использовать и другие поля Карточки поиска, например, поле «**Вид документа**».



- Рекомендации по заполнению поля **«Название документа»**: В поле укажите несколько слов, которые заведомо присутствуют в названии документа. Слова следует задавать полностью. Будут найдены документы, в названиях которых содержится эти слова во всех грамматических формах. Наряду с полными наименованиями сложно задавать их общепринятые сокращения и аббревиатуры, например: БУХУЧЁТ, НДС.

Если нужно получить информацию по правовому вопросу с использованием перечня возможных ситуаций

Найдем ответ на вопрос, как начислять амортизацию основных средств, бывших в употреблении:

- Нажмите кнопку **«Правовой навигатор»** в **Стартовом окне** системы;
- Наберите в строке поиска **«Амортизация ОС»**;
- В левом столбце установите курсор на названии группы **«Амортизация, износ»**;
- В правом столбце отметим галочкой ключевое понятие **«Основные средства, бывшие в употреблении»**;
- Нажмите кнопку **«Построить список документов (F9)»**;
- Получите дерево-список документов, содержащее нормативноправовые акты и разъяснения по искомому вопросу;
- Начните изучение вопроса с Налогового кодекса РФ (часть вторая), щелкните мышью по его названию;
- Перейдите в текст кодекса к помеченному фрагменту, который относится к выбранному ключевому понятию;
- С помощью кнопки **«Следующий фрагмент»** посмотрите другие фрагменты документа, соответствующие ключевому понятию;
- Повторно нажмите кнопку, чтобы перейти к следующему фрагменту.

Получение консультаций по вопросу – можно изучить остальные документы из списка найденных документов, в том числе консультации по рассматриваемому вопросу. Для этого выберите в дереве-списке **информационный банк Вопросы-ответы**.

Если надо получить консультации по применению конкретной статьи документа

Пример: Найдём разъяснение и консультации к статье 911 Гражданского кодекса РФ.

- Нажмите на кнопку **«Быстрый поиск»**;
- Зададим в строке Быстрого поиска: **«Статья 911 ГК РФ»**;
- Нажмите кнопку **«Найти»**;
- Откроется окно со списком документов, наиболее соответствующих запросу, Гражданский кодекс РФ (часть вторая) расположен первым в списке. Под названием кодекса размещена ссылка на искомую статью;
- Щелкните по ссылке, документ сразу откроется на статье 911 «Проверка качества и состояния товара при возвращении его товаровладельцу»;
- Щелкните мышью по значку [i] слева от названия статьи; откроется дерево-список документов, содержащих дополнительную информацию (консультации, судебную практику, комментарии) к фрагменту;
- Выберите в левой части дерева-списка информационный банк **«Постатейные комментарии и книги»** раздела **«Комментарии законодательства»**;

- Ознакомьтесь с представленными в правой части материалами. **Обзор изменений в законодательстве**

Пример: Найдём и изучаем обзор новых документов для бухгалтера. □

Нажмите кнопку **«Обзоры»**;

- В разделе **«Аналитические обзоры»** правовой информации щёлкните по ссылке **Новые документы** для бухгалтера;

- Откроется окно с результатами поиска, раздел содержит выпуски еженедельных образцов новых документов для бухгалтера;

- Откройте обзор, размещенный в начале списка;

- В обзоре рассмотрены наиболее интересные для бухгалтера документы с практическими рекомендациями по их применению;

- В верхней части документа представлено его содержание – перечень новостей и комментариев, откуда по ссылке можно перейти непосредственно к тексту интересной вам новости;

- Аналогичный перечень продублирован в оглавлении документа (кнопка **«Оглавление» Правой панели**).

Обзоры законодательства также содержат:

- Другие аналитические обзоры (обзор антикризисных документов, правовые новости (коротко о важном), изменения налогового законодательства с 2009 года, анонс журналов издательства «Главная книга» и пр.);

- Оперативные обзоры (ежедневный, еженедельный);

- Тематические подборки (материалы по составлению и предоставлению налоговой и бухгалтерской отчетности, по кадровым вопросам, банковскому делу, материалы для бюджетных организаций).

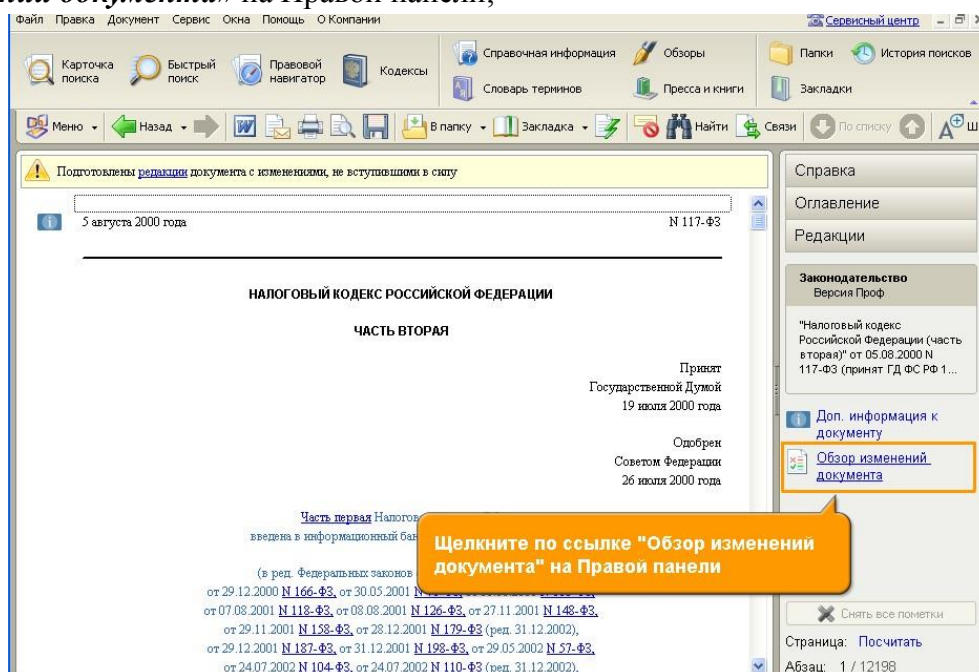
Если требуется изучить изменения в документе

Пример: Найдём и изучим обзор изменения части второй налогового кодекса РФ. Дополнительно найдём редакцию кодекса, действующую с 1 февраля 2009 года.

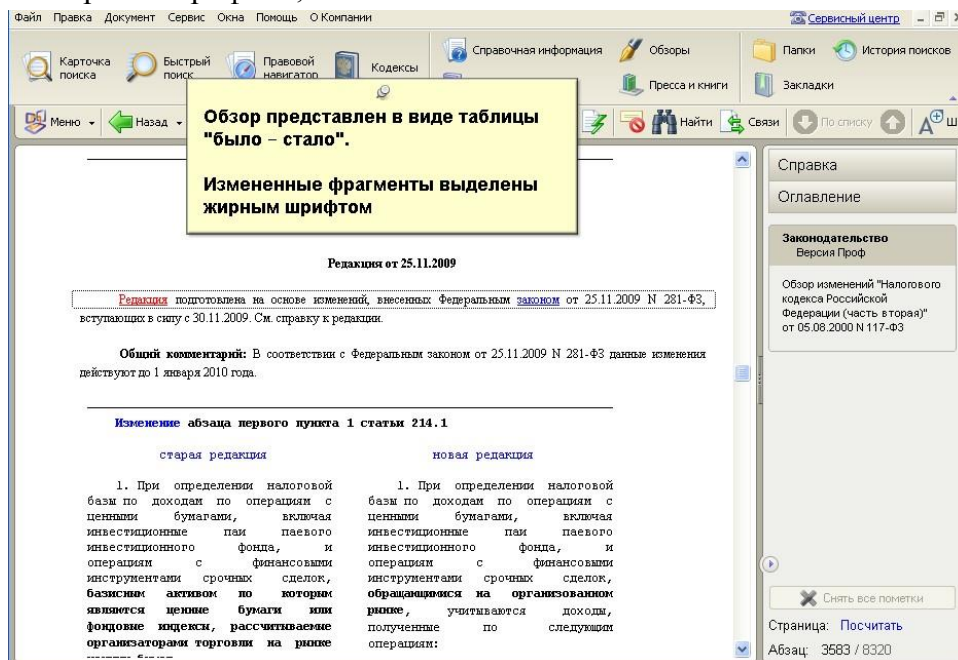
- Нажмите кнопку **«Кодексы»**;

- В появившемся списке найдите «Налоговый кодекс РФ (часть вторая)» и щёлкните мышью по его названию для перехода к текстовому документу;

- В открывшемся окне с текстовым документом щёлкните по ссылке **«Обзор изменений документа»** на Правой панели;



- Обзор представлен в виде таблице «*было - стало*», измененные фрагменты выделены жирным шрифтом;



Структура обзора изменения документа:

Обзор изменения документа построен в виде таблицы, содержащей измененные фрагменты документа: в левом столбце приводится текст фрагмента в старой редакции, в правом столбце – в новой редакции, причём изменения выделены жирным шрифтом. Если в документ вводится статья или пункт целиком, то даётся ссылка на этот пункт в тексте самого документа.

- Вернитесь в текст, нажав кнопку «**Назад**»;
- Откроется действующая редакция кодекса;
- Для поиска редакции, действовавшей на 01.02.2009, нажмите кнопку «**Редакции**» на Правой панели;
- В окне «**Редакции**» наберите в строке поиска дату «**01.02.2009**» и нажмите кнопку «**Найти**»;
- Откроется текст редакции документа, действовавшей на указанную дату.

Как определить период действия редакции:

- Следует перейти к справкам к этой редакции, кликнув мышью по кнопке «**Справка**» на **Правой панели**.

Практическая работа №8.

Создание контекстной диаграммы с помощью Ramus educational.

Цель: Ознакомиться с инструментальной средой Ramuseducational.

Изучить основные объекты диаграмм. Научиться строить контекстные диаграммы, диаграммы декомпозиции и отчёты.

В результате выполнения практических занятий обучающийся должен **уметь**: □ выделять жизненные циклы проектирования информационной системы;

- использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;

В результате выполнения практических занятий обучающийся должен **знать**:

- цели автоматизации производства;

- типы организационных структур;
- требования к проектируемой системе;
- классификацию информационных систем, структуру информационной системы, понятие жизненного цикла информационной системы;
- модели жизненного цикла информационной системы, методы проектирования информационной системы;
- оценку необходимых ресурсов для реализации проекта.

Теоретическая часть

Создание современного программного обеспечения для информационных систем представляет собой сложнейшую задачу, решение которой требует применения специальных методик и инструментов. Неудивительно, что в последнее время среди системных аналитиков и разработчиков значительно вырос интерес к *CASE - технологиям* и инструментальным *CASE-средствам*, позволяющим максимально систематизировать и автоматизировать все этапы разработки программного обеспечения.

CASE-технологии представляют собой совокупность методологий и инструментарий аналитиков, разработчиков и программистов, предназначенный для автоматизации процессов проектирования и сопровождения АС на всем ее жизненном цикле.

CASE-средства - это средства, помогающие разрабатывать сложное ПО с помощью компьютера.

При разработке программного обеспечения для ИС необходимо, прежде всего, проанализировать предметную область, т.е. выявить все бизнес-процессы.

Ramuseducational относится к малым интегрированным средствам моделирования, которые поддерживают несколько типов моделей и методов. BPwin предназначен для проведения анализа и реорганизации бизнес-процессов.

Ramuseducational автоматизирует задачи, связанные с построением моделей развития, обеспечивая семантическую строгость, необходимую для гарантирования правильности и непротиворечивости результатов.

Ramuseducational поддерживает три методологии: IDEF0, DFD и IDEF3:

В рамках методологии **IDEF0** (Integration Definition for Function Modeling) бизнес-процесс представляется в виде набора элементов-работ, которые взаимодействуют между собой, а также показывается информационные, людские и производственные ресурсы, потребляемые каждой работой.

Диаграммы **DFD** (Data Flow Diagramming) могут дополнить то, что уже отражено в модели IDEF3, поскольку они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией между бизнес-функциями внутри системы. В тоже время диаграммы DFD оставляют без внимания взаимодействие между бизнес-функциями.

IDEF3. Этот метод привлекает внимание к очередности выполнения событий. В IDEF3 включены элементы логики, что позволяет моделировать и анализировать альтернативные сценарии развития бизнес-процесса.

BPwin умеет проверять создаваемые модели с точки зрения синтаксиса выбранной методологии, проверяет ссылочную целостность между диаграммами, а также выполняет ряд других проверок, чтобы помочь вам создать правильную модель, а не просто рисунок. При этом сохраняются главные преимущества рисунка - простота создания и наглядность. IDEF0.

Контекстная диаграмма - это модель, которая представляет систему как набор действий, в которые каждое действие преобразует некоторый объект или набор объектов. Модель представляется как набор иерархических действий. Высшее действие иерархии называется действием контекста. Это самый высокий уровень, который непосредственно описывает систему. Уровни ниже называются порожденными декомпозициями и представляют под процессы родительского действия.

При создании модели сначала необходимо изобразить самый высокий уровень, действие контекста. Наименование действия описывает систему непосредственно и, как правило, состоит из одного активного глагола в сочетании с обобщающим существительным, которое разъясняет цель деятельности с точки зрения самого общего взгляда на систему.

Двумя наиболее важными компонентами, из которых строятся диаграммы IDEF0, являются **бизнес-функции или работы**(представленные на диаграммах в виде прямоугольников) и **данные и объекты** (изображаемые в виде стрелок), связывающие между собой работы. При этом стрелки, в зависимости от того в какую грань прямоугольника работы они входят или из какой грани выходят, делятся на пять видов:

Стрелки входа (входят в левую грань работы) - изображают данные или объекты, изменяемые в ходе выполнения работы.

Стрелки управления (входят в верхнюю грань работы) - изображают правила и ограничения, согласно которым выполняется работа.

Стрелки выхода (выходят из правой грани работы) - изображают данные или объекты, появляющиеся в результате выполнения работы.

Стрелки механизма (входят в нижнюю грань работы) - изображают ресурсы, необходимые для выполнения работы, но не изменяющиеся в процессе работы (например, оборудование, людские ресурсы...).

Стрелки вызова (выходят из нижней грани работы) - изображают связи между разными диаграммами или моделями, указывая на некоторую диаграмму, где данная работа рассмотрена более подробно.

Все работы и стрелки должны быть именованы. Первая диаграмма в иерархии диаграмм IDEF0 всегда изображает функционирование системы в целом. Такие диаграммы называются контекстными. В контекст входит описание цели моделирования, области (описания того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и точки зрения (позиции, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственного за работу моделируемой системы в целом.

Контекстная диаграмма изображает деятельность самого верхнего уровня и обозначает границу моделирования относительно цели, возможностей и точки зрения. Название контекстной диаграммы находится в дереве модели непосредственно под общим описанием.

Декомпозиционное разложение модели используется в моделировании бизнес-процессов, чтобы дать более подробное описание блоков. Каждое из этих действий может в свою очередь быть декомпозировано. При каждой декомпозиции блока создается новая диаграмма. Число декомпозиций не ограничено и полностью зависит от уровня сложности, который необходимо показать в модели.

Диаграммы декомпозиции предназначены для детализации функций и получаются при разбиении контекстной диаграммы на крупные подсистемы (функциональная декомпозиция) и описывающие каждый подсистему и их взаимодействие.

Единственная **функция**, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания дочерней диаграммы. В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания дочерней диаграммы следующего, более низкого уровня, на которой некоторые или все функции также могут быть разложены на составные части. Каждая **дочерняя диаграмма** содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

Хотя действительной вершиной модели является диаграмма уровня A-0, настоящей «рабочей вершиной» является диаграмма A0, поскольку она является уточненным выражением точки зрения модели. Ее содержание показывает, что будет рассматриваться в дальнейшем, ограничивая последующие уровни в рамках цели проекта. Нижние уровни уточняют содержание функциональных блоков, детализируя их, но, не расширяя границ модели.

Практическая работа №9.

Создание диаграмм декомпозиции с помощью Ramus educational.

Цель: Ознакомиться с инструментальной средой Ramuseducational. Изучить основные объекты диаграмм. Научиться строить контекстные диаграммы, диаграммы декомпозиции и отчёты.

Практическая часть

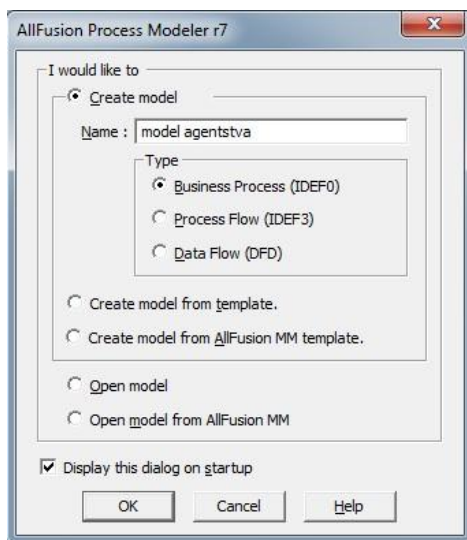
Задание 1: Создать контекстную диаграмму и отчёты, согласно варианту задания.

Задание 2: На основе, построенной контекстной диаграммы, создать диаграмму декомпозиции.

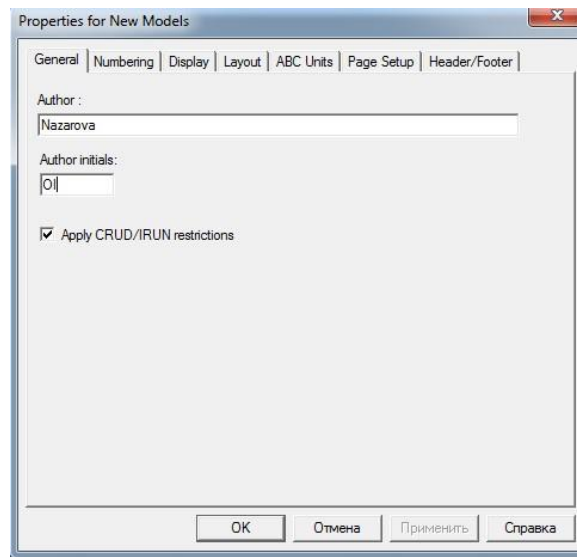
Задание 3: Ответить на контрольные вопросы, оформить отчёт.

Создание контекстной диаграммы

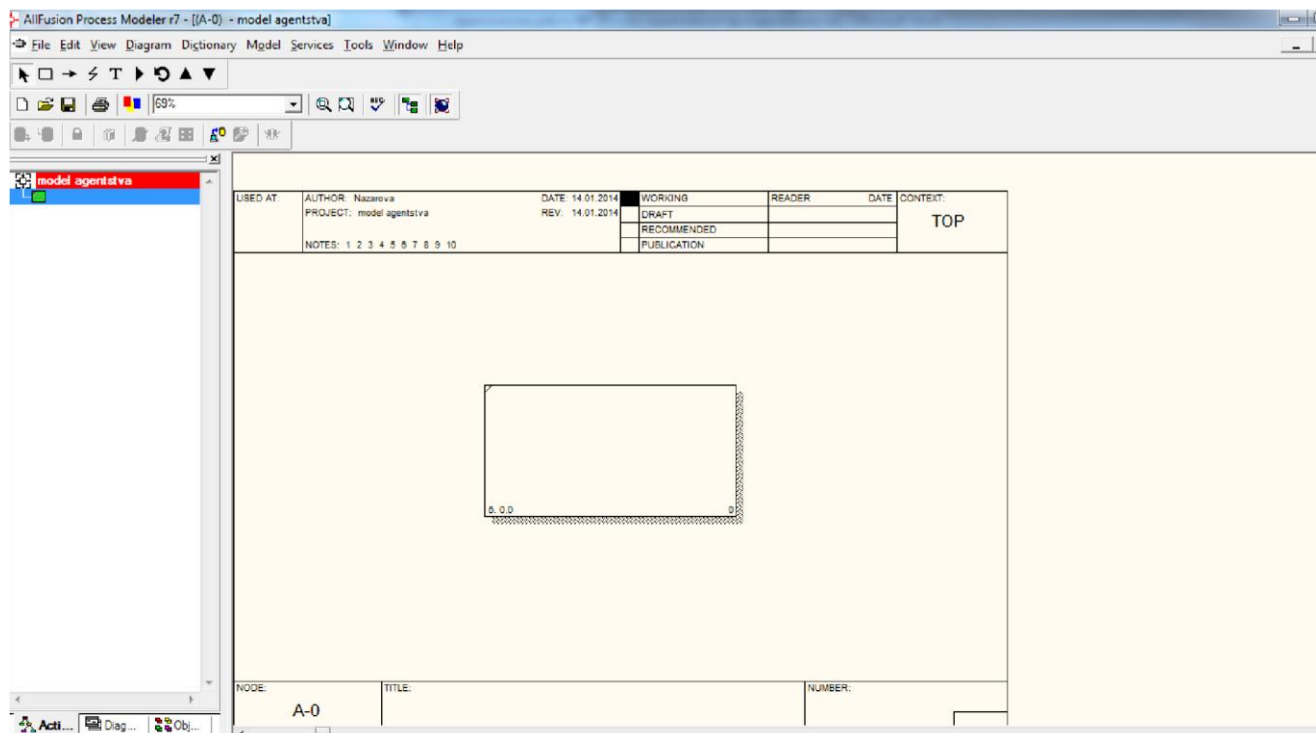
1. Для того чтобы создать контекстную диаграмму запустите программу Ramuseducational. Появится диалоговое окно, в котором необходимо указать **Имя модели** и **тип диаграммы**.



2. Далее в диалоговом окне **PropertiesforNewModels** (Свойства новой модели) указываем имя автора и его инициалы.



3. Создаётся незаполненная контекстная диаграмма



4. Инструмент просмотра и навигации **ModelExplorer (Браузер модели)** имеет три вкладки – **Activities, Diagrams, Objects**.



Выбрав вкладку **Activities** щелчком правой кнопки по объекту в браузере модели можно выбрать операции редактирования его свойств.

5. Чтобы указать **имя модели** и **имя проекта** необходимо перейти в меню **Model/Model Properties** на вкладку **General**

Model Properties

Layout | ABC Units | Page Setup | Header/Footer | Shapes | Draw Style
General | Purpose | Definition | Source | Status | Numbering | Display

Model name:
model agentstva

Project:
model proectirovaniya technologicheskix processov agentstva

Author :
Nazarova

Author initials:
OI ☒ Apply CRUD/IRUN restrictions

Time Frame
☒ AS-IS
☐ TO-BE

OK Отмена Применить Справка

6. Во вкладке **Purpose** диалогового окна **Model Properties** необходимо внести данные о цели разработки модели (**Purpose**) и точку зрения (**Viewpoint**).

Model Properties

Layout | ABC Units | Page Setup | Header/Footer | Shapes | Draw Style
General | Purpose | Definition | Source | Status | Numbering | Display

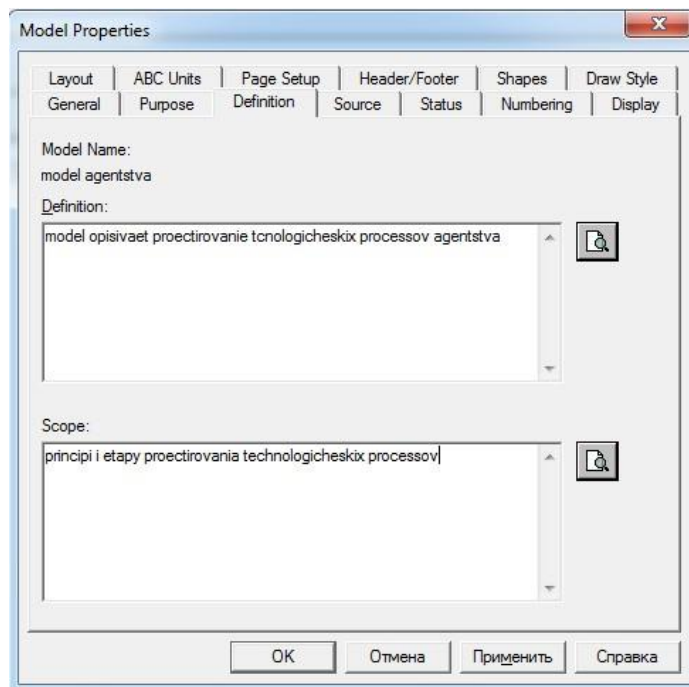
Model Name:
model agentstva

Purpose:
Razrabotat posledovatelnost i processi proectirovaniya

Viewpoint:
director

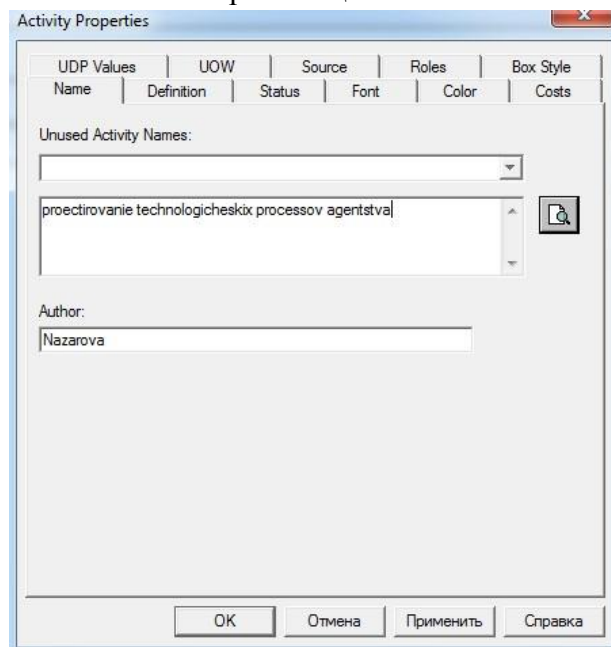
OK Отмена Применить Справка

7. Во вкладке **Definition** диалогового окна **Model Properties** ввести определение модели (**Definition**) и охват процессов модели (**Scope**)



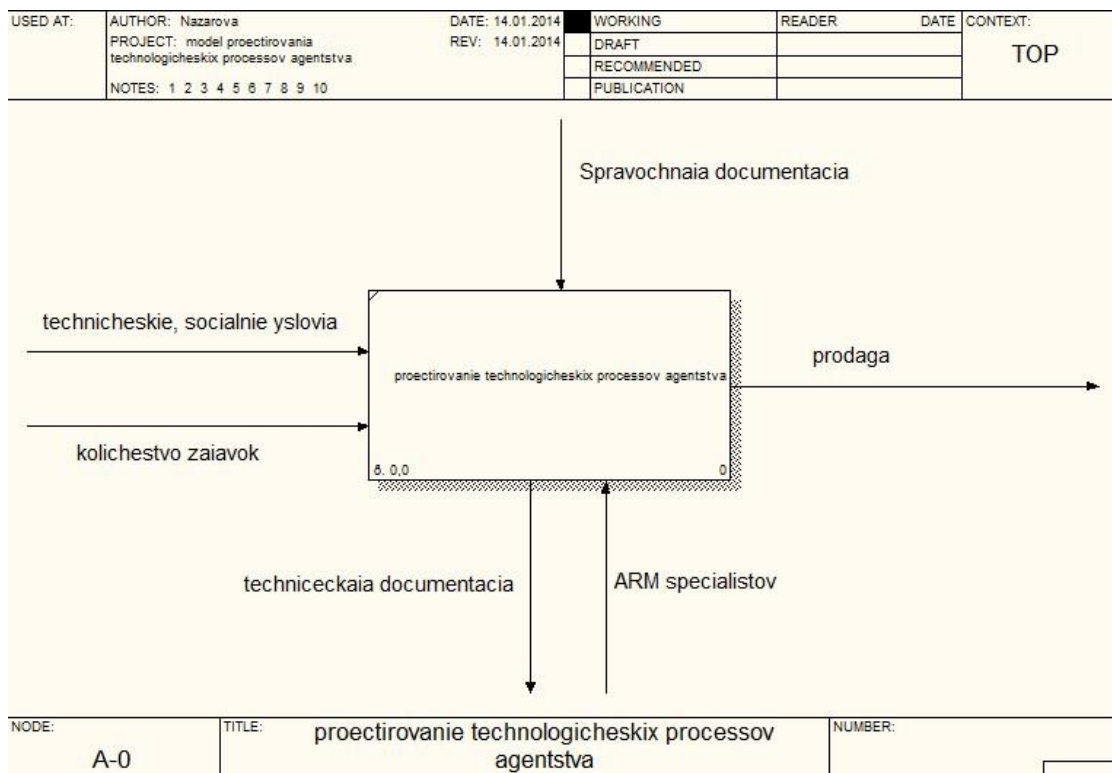
8. **Работы** на контекстной диаграмме отображаются прямоугольниками и обозначают поименованные процессы, функции или задачи, которые происходят в течение определённого времени и имеют распознаваемые результаты.

Чтобы внести имя **Работы** необходимо правой кнопкой мыши щелкнуть по прямоугольнику, в контекстном меню выбрать опцию **Name**.



10. Взаимодействие работ с внешним миром и между собой описывается в виде **стрелок**.

Для полной картины контекстной диаграммы необходимо создать различные типы стрелок и внести текст в поле диаграммы.



12. BPwin имеет мощный инструмент генерации отчетов. Отчеты по модели вызываются из пункта меню **Report**. Всего имеется семь типов отчетов:

1. **Model Report.** Этот отчет включает информацию о контексте модели - имя модели, точку зрения, область, цель, имя автора, дату создания и др.
2. **Diagram Report.** Отчет по конкретной диаграмме. Включает список объектов (работ, стрелок, хранилищ данных, внешних ссылок и т.д.).
3. **Diagram Object Report.** Наиболее полный отчет по модели. Может включать полный список объектов модели (работ, стрелок с указанием их типа и др.) и свойства, определяемые пользователем.
4. **Activity Cost Report.** Отчет о результатах стоимостного анализа.
5. **Arrow Report.** Отчет по стрелкам. Может содержать информацию из словаря стрелок, информацию о работе-источнике, работеназначении стрелки и информацию о разветвлении и слиянии стрелок.
6. **Data Usage Report.** Отчет о результатах связывания модели процессов и модели данных.
7. **Model Consistency Report.** Отчет, содержащий список синтаксических ошибок модели.

Практическое занятие №10. Диаграммы декомпозиции.

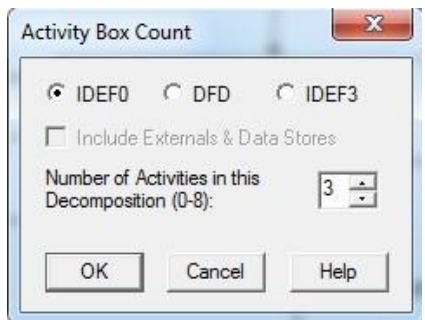
Цель: Ознакомиться с инструментальной средой Ramus educational.
Изучить основные объекты диаграмм. Научиться строить контекстные диаграммы, диаграммы декомпозиции и отчёты.

▼

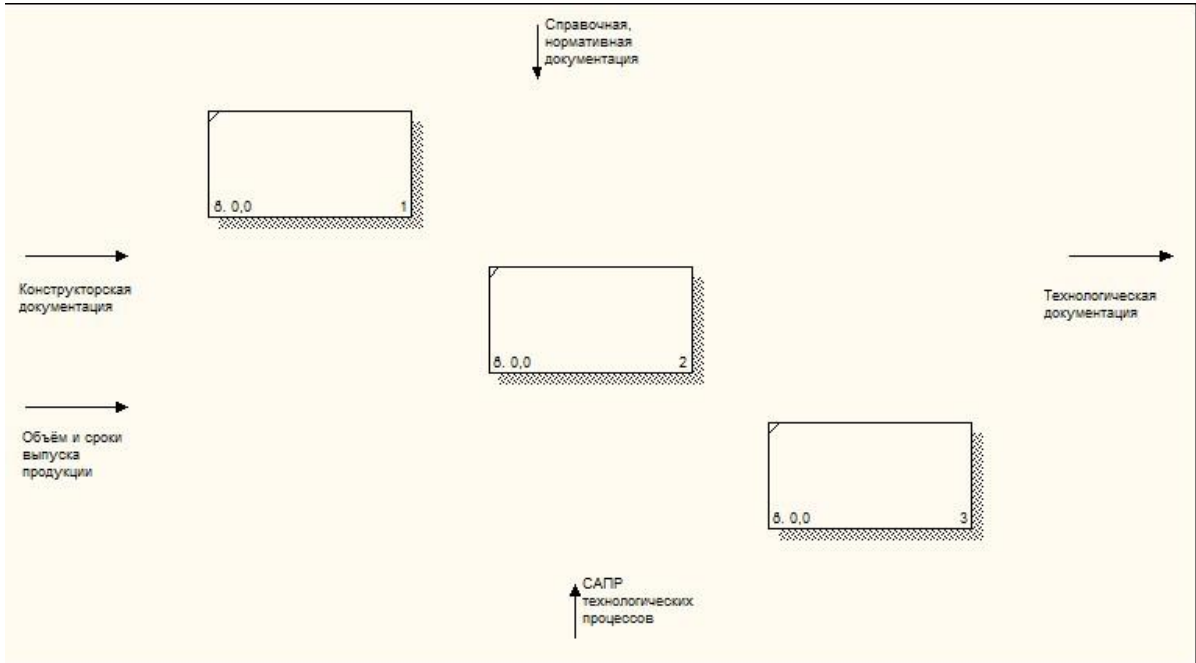
Создание диаграммы декомпозиции 1. Выберите кнопку

перехода на нижний уровень в палитре инструментов и в диалоговом окне

ActivityBoxCount установите число работ на диаграмме нижнего уровня – 3 – и нажмите кнопку **OK**.



2. Автоматически будет создана диаграмма декомпозиции



Правой кнопкой мыши щелкните по работе, расположенной в левом верхнем углу области редактирования модели, выберите в контекстном меню опцию **Name** и ввести имя работы. Повторите операцию для оставшихся двух работ. Затем внести определение, статус и источник для каждой работы.

Таблица 1 – Работы и их характеристики

Название работы (Activity Name)	Определение работы (Activity Definition)
Анализ	Анализ исходных данных, объёма выпуска продукции, определение типа производства и составление технологического процесса.
Проектирование	Разработка технологических операций, расчёт технико-экономической эффективности.
Документировани е	Оформление технологической документации, разработка технического задания.

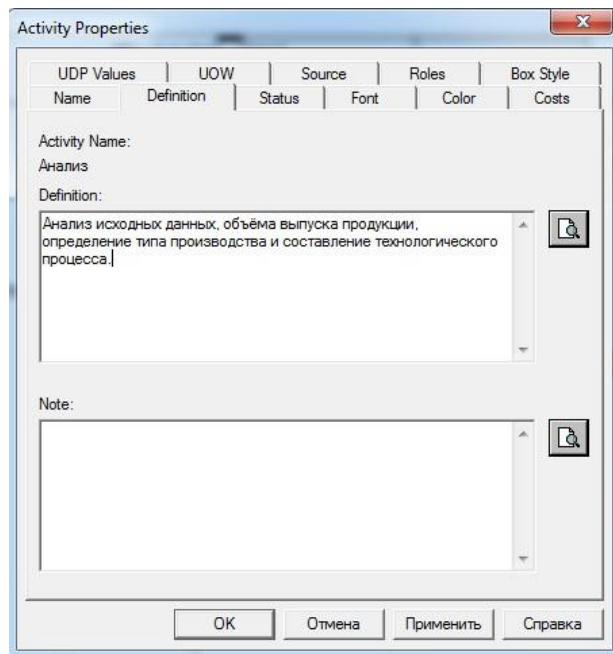
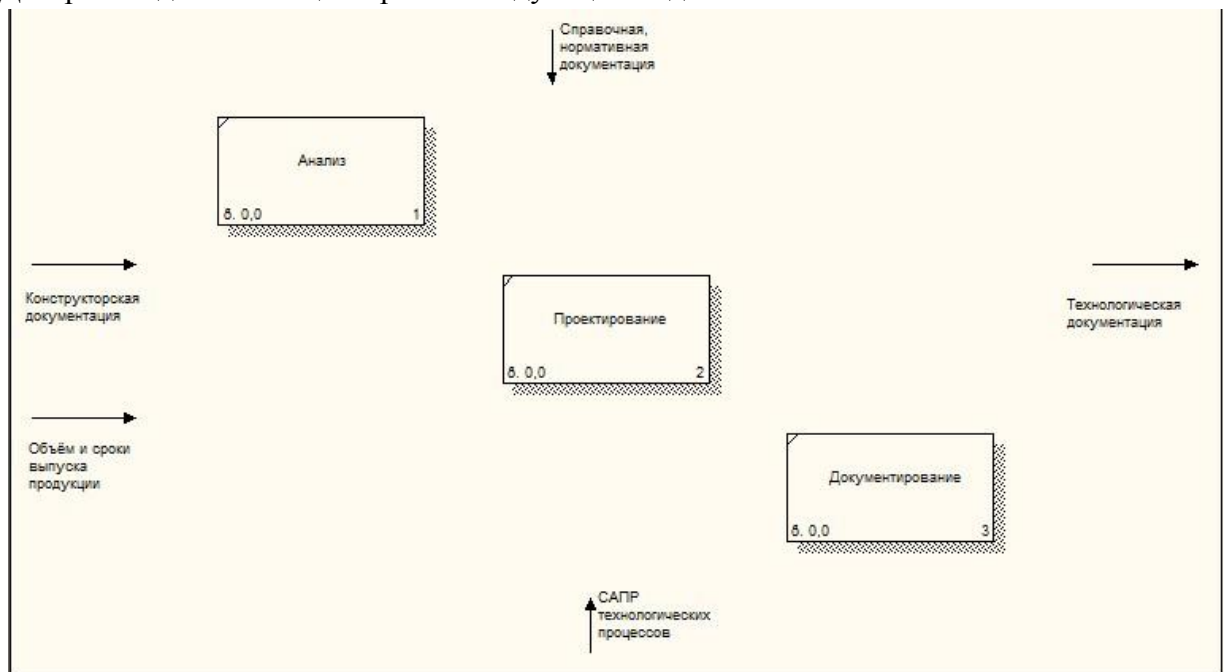


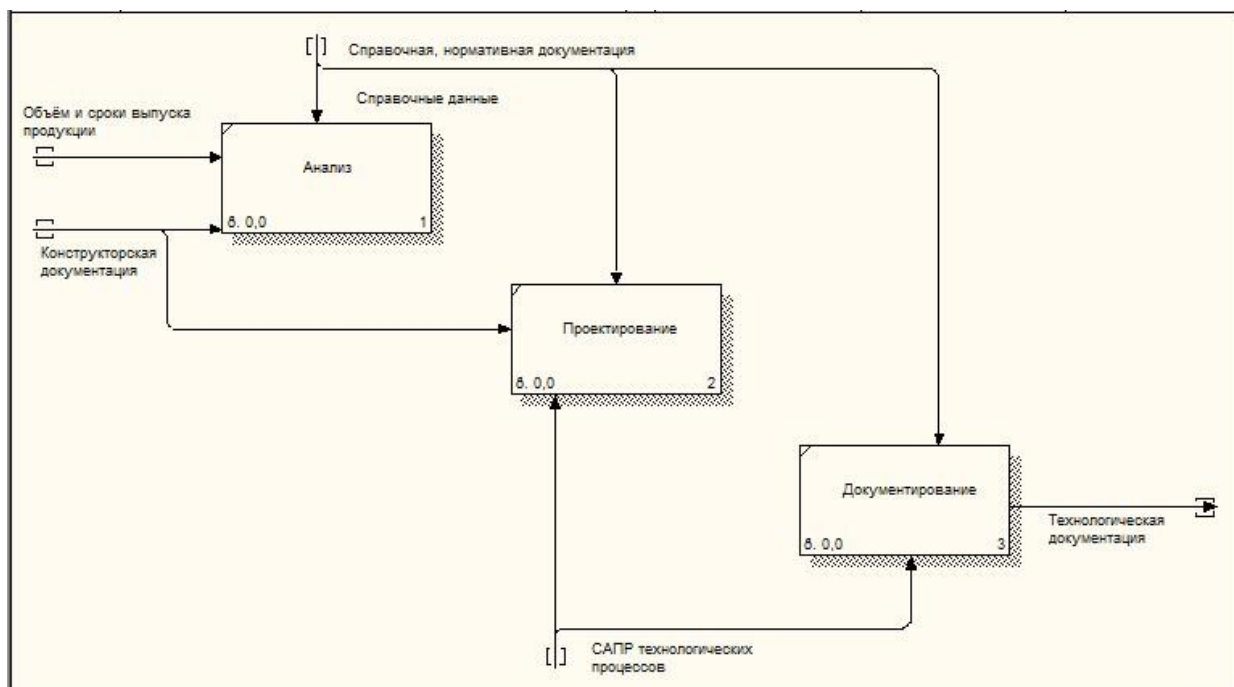
Диаграмма декомпозиции примет следующий вид



Для изменения свойств работ после их внесения в диаграмму можно воспользоваться словарем работ. Вызов словаря производится при помощи пункта главного меню **Dictionary/Activity**.

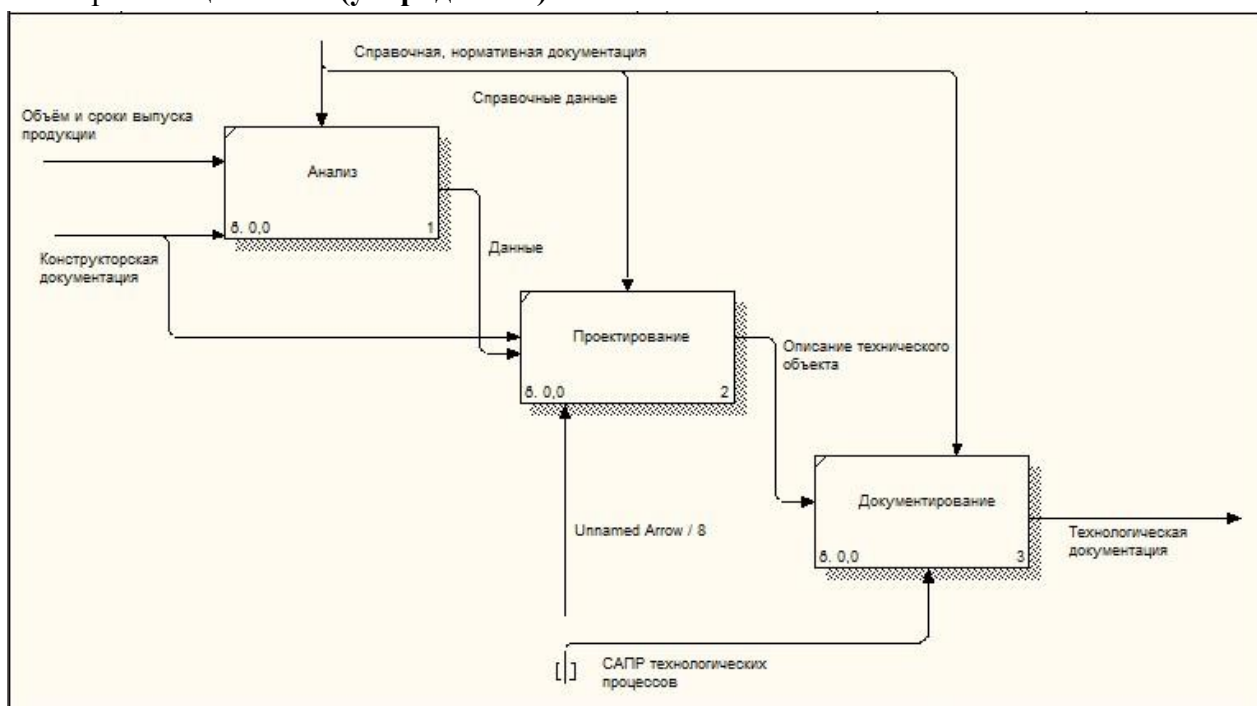
3. Перейдите в режим рисования стрелок и свяжите граничные стрелки, воспользовавшись кнопкой  на палитре инструментов.

4. Правой кнопкой мыши щелкните по ветви стрелки управления работы **Справочная, нормативная документация** и переименовать в **Справочные данные**.

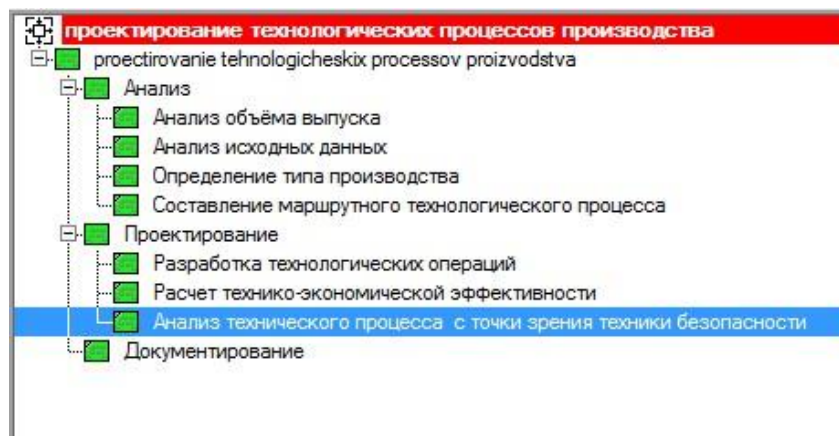


5. Создайте новые внутренние стрелки

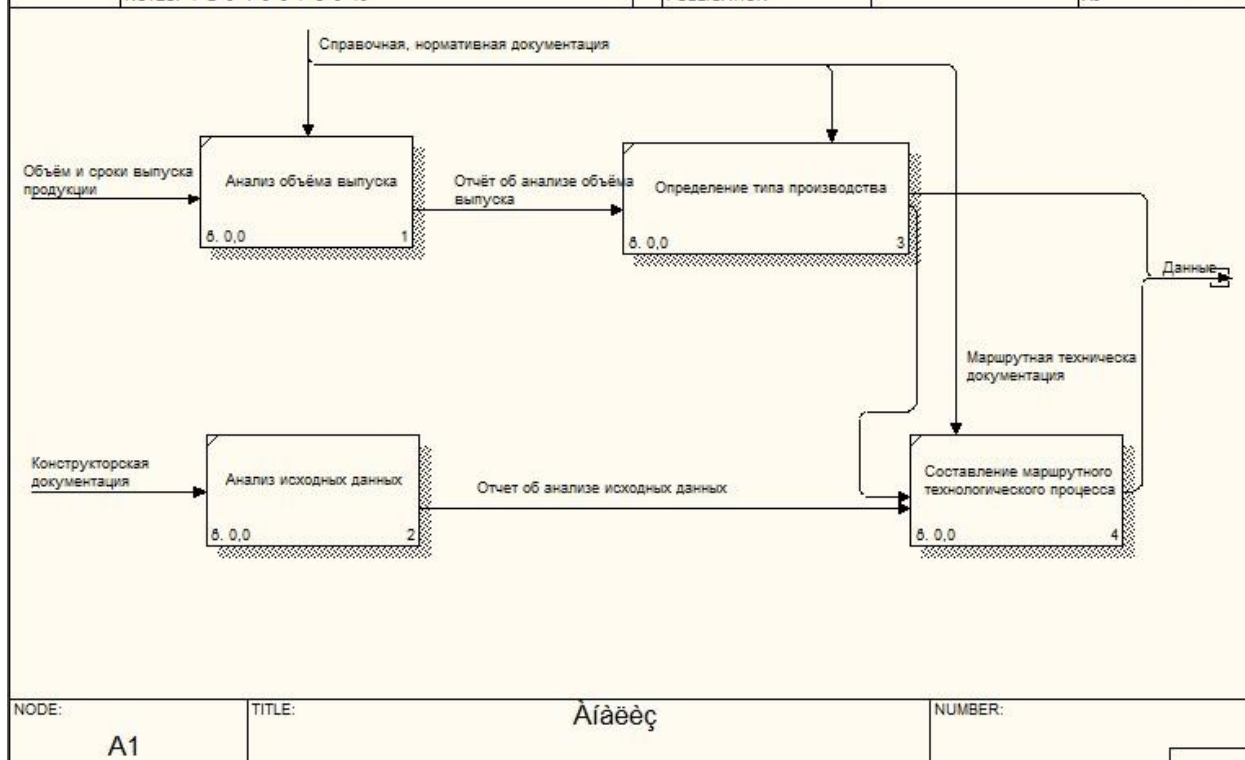
Граничные стрелки, которые имеют вид с квадратными скобками на наконечнике  автоматически не попадают на диаграмму верхнего уровня. Щелкните правой кнопкой мыши по квадратным скобкам и выберите пункт меню **ArrowTunnel**. В диалоговом окне **BorderArrowEditor (редактор граничных стрелок)** выберите опцию **ResolveittoBorderArrow (разрешить как граничную стрелку)**. Данная стрелка попадет на диаграмму верхнего уровня, а квадратные скобки пропадут. Для этой граничной стрелки выберите опцию **Trim (упорядочить)** из контекстного меню.

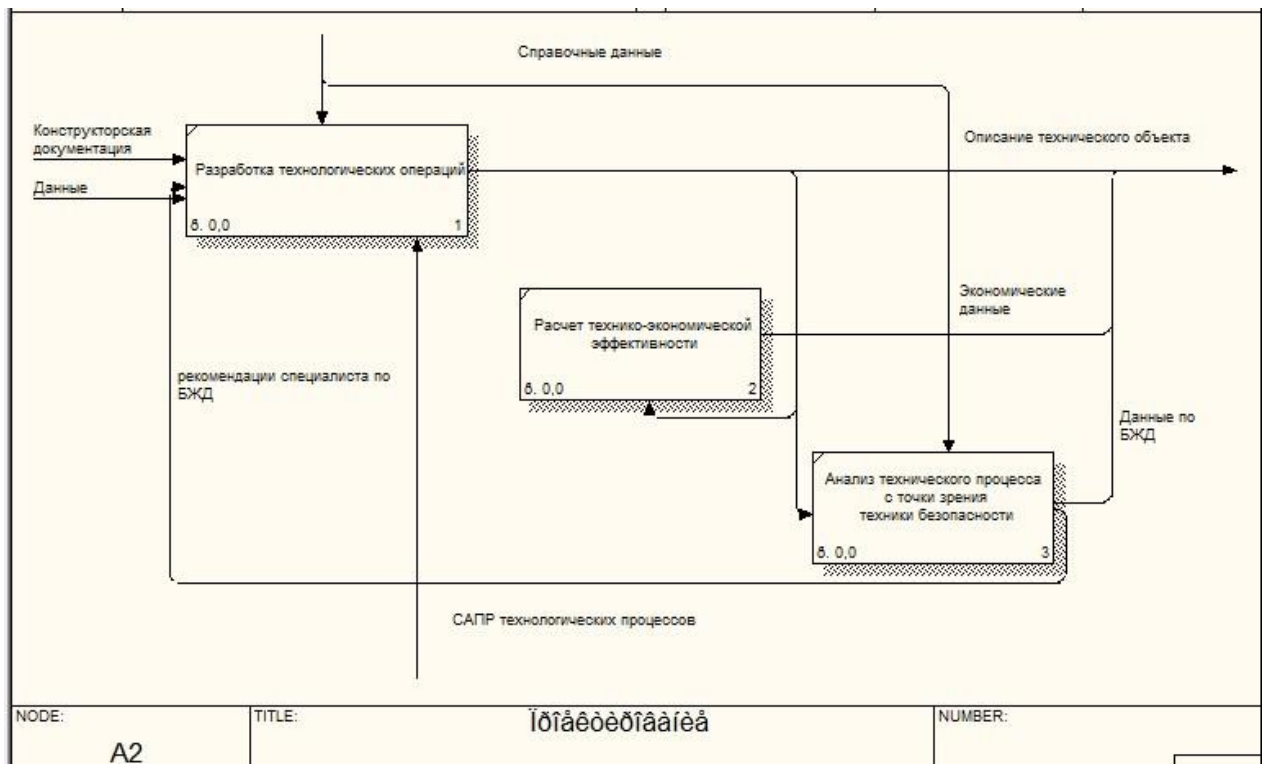


6. Декомпозируйте каждую новую работу



USED AT:	AUTHOR: Nazarova	DATE: 21.01.2014	WORKING	READER	DATE	CONTEXT:
	PROJECT: model proizvodstva	REV: 21.01.2014	DRAFT			
			RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10		PUBLICATION			A0





Варианты предметных областей

Таблица 2 – Варианты предметных областей

№ варианта	Предметная область
1	Деятельность автосалона
2	Деятельность автозаправочного комплекса
3	Деятельность АТС
4	Деятельность службы знакомств
5	Деятельность отдела кадров
6	Деятельность автотранспортного предприятия
7	Деятельность предприятия по производству мебели
8	Деятельность диспетчерской службы такси
9	Деятельность железнодорожных касс
10	Деятельность магазина бытовой техники
11	Деятельность фирмы по организации торжеств
12	Деятельность коммерческого банка
13	Деятельность интернет-кафе
14	Деятельность военкомата
15	Деятельность инвентаризационной фирмы
16	Деятельность отдела сбыта
17	Деятельность агентства недвижимости
18	Деятельность службы занятости
19	Деятельность аптеки
20	Деятельность базы отдыха
21	Деятельность фирмы по ремонту и сборке компьютерной техники
22	Деятельность музея
23	Деятельность отдела снабжения предприятия
24	Деятельность авиакасс
25	Деятельность гостиницы
26	Деятельность печатного издательства
27	Деятельность предприятия оптовой торговли лекарственными препаратами

Контрольные вопросы:

1. Дайте определения Case-технологиям и Case-средства;
2. Методологии поддерживаемые Ramuseducational;
3. Назовите виды стрелок в Ramuseducational;
4. Перечислите типы отчётов Ramuseducational.
5. Что такое Ramuseducational;
6. Задачи, решаемые при помощи Ramuseducational;
7. Что такое контекстная диаграмма;
8. Что такое диаграмма декомпозиции. Для чего используют декомпозирование работ.

Практическая работа №11.

Создание диаграммы дерева узлов и диаграммы потоков работ (IDEF3).

Цель: Изучить основные определения и понятия методологии IDEF3. Познакомиться с основными способами построения диаграммы дерева узлов. Научиться строить диаграммы потоков работ и диаграммы дерева узлов.

В результате освоения учебной дисциплины обучающийся должен **уметь**:

- ☐ выделять жизненные циклы проектирования информационной системы;
- ☐ использовать методы и критерии оценивания предметной области и методы определения стратегий развития бизнес-процессов организации;

В результате освоения учебной дисциплины обучающийся должен **знать**:

- ☐ типы организационных структур;
- ☐ требования к проектируемой системе;
- ☐ модели жизненного цикла информационной системы, методы проектирования информационной системы;
- ☐ технологии проектирования информационной системы, оценку и управление качеством информационной системы;
- ☐ организацию труда при разработке информационной системы; ☐ оценку необходимых ресурсов для реализации проекта.

Теоретическая часть

Дерево узлов - представление отношений между родительскими и дочерними узлами модели IDEF0 в форме древовидного графа. **Диаграммы дерева узлов** – это диаграммы, показывающие не взаимосвязи между функциями (стрелки), а иерархическую зависимость функций. Диаграмма узлов использует традиционное дерево иерархий, в котором верхний узел (блок) соответствует контекстной диаграмме, а нижний уровень – декомпозицию потомков.

Имя дерева узлов по умолчанию совпадает с именем блока верхнего уровня, а номер диаграммы автоматически генерируется как номер узла верхнего уровня плюс буква «N», например A0N. Если в модели создается два дерева узлов, имеющие в качестве верхнего уровня одну и ту же функцию, то по умолчанию диаграммы получают идентичные номер и имя. Поэтому рекомендуется при **создании диаграммы** дерева узлов задавать имя диаграммы, отличное от значения по умолчанию.

При создании дерева узлов обязательно указывается имя диаграммы, т.к. если в нескольких диаграммах в качестве корня на дереве узлов использовать одну и ту же функцию, то все эти диаграммы будут иметь одинаковый номер (номер узла + постфикс N, например A0N). В этом случае их можно будет различить по имени.

Диаграммы потоков работ

Наличие в диаграммах DFD элементов для описания источников, приемников и хранилищ данных позволяет более эффективно и наглядно описать процесс документооборота. Однако для описания логики взаимодействия информационных потоков более подходит **IDEF3**, называемая также workflow diagramming – методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. **Диаграммы Workflow** могут быть использованы в

моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 – это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа.

Каждая работа в IDEF3 описывает какой-либо сценарий бизнеспроцесса и может являться составляющей другой работы. Поскольку сценарий описывает цель и рамки модели, важно, чтобы работы именовались отглагольным существительным, обозначающим процесс действия, или фразой, содержащей такое существительное.

Диаграммы. Диаграмма является основной единицей описания в IDEF3. Важно правильно построить диаграммы, поскольку они предназначены для чтения другими людьми (а не только автором).

Единицы работы – Unit of Work (UOW). UOW, также называемые работами (activity), являются центральными компонентами модели. В IDEF3 работы изображаются прямоугольниками с прямыми углами и имеют имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор); другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы (например, «*Изготовление изделия*»). Обычно номер работы состоит из номера родительской работы и порядкового номера на текущей диаграмме.

Связи. Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправленные и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо. В IDEF3 различают три типа стрелок, изображающих связи, стиль которых устанавливается через меню **Edit/Arrow Style**:

Старшая (Precedence) – сплошная линия, связывающая единицы работ (UOW). Рисуются слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

Отношения (Relational Link) – пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок.

Потоки объектов (Object Flow) – стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

Старшая связь и поток объектов. Старшая связь показывает, что работа-источник заканчивается ранее, чем начинается работа-цель. Часто результатом работы-источника становится объект, необходимый для запуска работы-цели. В этом случае стрелку, обозначающую объект, изображают с двойным наконечником. Имя стрелки должно ясно идентифицировать отображаемый объект. Поток объектов имеет ту же семантику, что и старшая стрелка.

Отношение показывает, что стрелка является альтернативой старшей стрелке или потоку объектов в смысле задания последовательности выполнения работ – работа-

источник не обязательно должна закончиться, прежде чем работа-цель начнется. Более того, работа-цель может закончиться прежде, чем закончится работа-источник



Рис. 1 Временная диаграмма выполнения работ

Перекрестки (Junction). Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления. Для внесения перекрестка служит кнопка – (добавить в диаграмму перекресток – Junction) в палитре инструментов. В диалоге Junction Type Editor необходимо указать тип перекрестка.

Смысл каждого типа приведен в таблице 1.

Таблица 1 – Типы перекрестков.

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Synchronous OR	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс J. Можно редактировать свойства перекрестка при помощи диалога Definition Editor. В отличие от IDEF0 и DFD в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Объект ссылки. Объект ссылки в IDEF3 выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой. Для внесения объекта ссылки служит кнопка (добавить в диаграмму объект ссылки – Referent) в палитре инструментов. **Объект ссылки изображается** в виде прямоугольника, похожего

на прямоугольник работы. Имя объекта ссылки задается в диалоге Referent (пункт всплывающего меню Name Editor), в качестве имени можно использовать имя какой-либо стрелки с других диаграмм или имя сущности из модели данных. Объекты ссылки должны быть связаны с единицами работ или перекрестками пунктирными линиями.

Официальная спецификация

IDEF3 различает три стиля объектов ссылок – безусловные (unconditional), синхронные (synchronous) и асинхронные (asynchronous).

ВРwin поддерживает только безусловные объекты ссылок. Синхронные и асинхронные объекты ссылок, используемые в диаграммах переходов состояний объектов, не поддерживаются.

Декомпозиция работ. В IDEF3 декомпозиция используется для детализации работ. Методология IDEF3 позволяет декомпозировать работу многократно, т. е. работа может иметь множество дочерних работ. Это позволяет в одной модели описать альтернативные потоки. Возможность множественной декомпозиции предъявляет дополнительные требования к нумерации работ. Так, номер работы состоит из номера родительской работы, версии декомпозиции и собственного номера работы на текущей диаграмме.

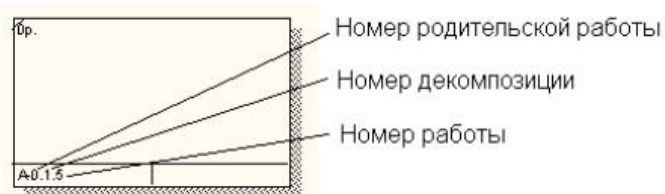


Рис.2 Определение номера работы

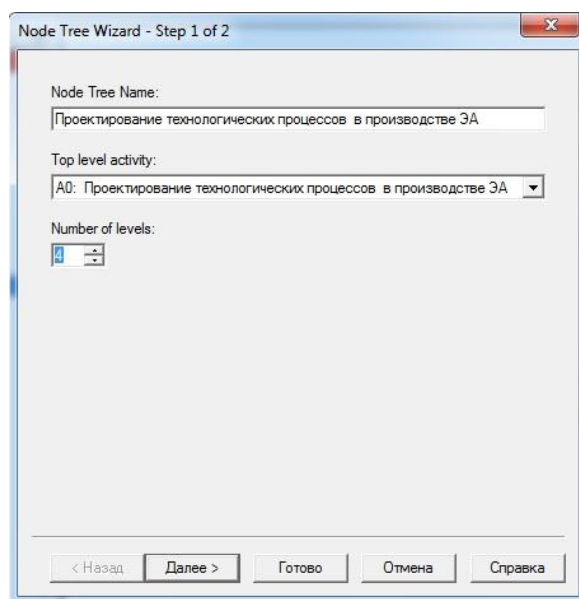
Практическая работа №12.
Ramus educational для создания диаграммы дерева узлов и диаграммы потоков работ (IDEF3).

Цель: Изучить основные определения и понятия методологии IDEF3. Познакомиться с основными способами построения диаграммы дерева узлов. Научиться строить диаграммы потоков работ и диаграммы дерева узлов.

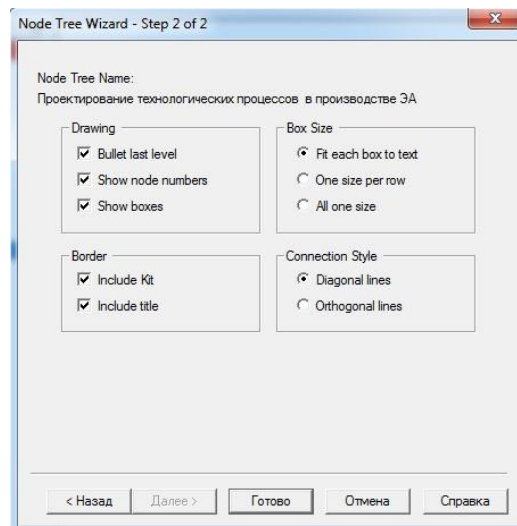
Практическая часть

Задание: На основе, построенных в предыдущих практических работах контекстной диаграммы и диаграмм декомпозиций, создать диаграмму дерева узлов и диаграммы потоков работ. **Пример создания диаграммы дерева узлов.**

Для создания диаграммы узлов в модели с помощью мастера необходимо выполнить команду **Diagram / Add Node Tree** . В открывшемся диалоговом окне **NodeTree Node Tree nameDiagram Wizard – Step 1 of 2** осуществляется настройка следующих опций:



- **Node Tree name** – текстовое поле в котором задается уникальное имя диаграммы дерева узлов, или пользователь может согласиться с предложенным именем по умолчанию. Имя по умолчанию соответствует имени диаграммы верхнего уровня в модели. Заданное имя диаграммы указывается в ячейке TITLE в заголовке диаграммы
 - **Top level activity** – список, в котором выбирается имя функции верхнего уровня в диаграмме. Поскольку дерево узлов не обязательно в качестве верхнего уровня должно иметь контекстный блок, то аналитик может поместить на верхний уровень диаграммы дерева узлов любую функцию. Такая диаграмма помогает облегчить понимание читателей диаграммы ее содержания.
 - **Number of levels** - список, в котором задается число уровней в диаграмме дерева узлов (глубина декомпозиции). Каждый уровень диаграммы соответствует уровню декомпозиции.
 - **Back** – кнопка перехода к предыдущему диалоговому окну мастера
 - **Next** - кнопка перехода к следующему диалоговому окну
 - **Finish** – кнопка закрытия диалогового окна и добавление диаграммы узлов
- Диалоговое окно **Node Tree Diagram Wizard – Step 2 of 2** содержит опции, определяющие свойства различных компонентов диаграммы дерева узлов:



- **Drawing** – зона, в которой содержатся переключатели:

☐ **Bullet last level** – изменение изображения функций нижнего уровня в виде прямоугольника на изображение в виде окружности. Применение изображения в виде окружности позволяет уменьшить пространство, занимаемое изображением функций нижнего уровня, особенно в тех случаях, когда на нижнем уровне используется большое количество

☐ **Show node numbers** – указание номеров узлов в нижнем правом углу изображения узла

☐ **Show boxes** – вычерчивание блоков для каждого узла ☐ **Box Size** - зона, в которой содержатся переключатели:

☐ **Fit each box to text** – создание прямоугольника вокруг надписи с именем узла. Размеры создаваемого прямоугольника зависят от длины надписи, поэтому применение такой опции может привести к наличию прямоугольников разных размеров

☐ **One size per row** – вычерчивание всех прямоугольников в одну строчку с размерами, равными размеру самого большого прямоугольника в строчке

☐ **All one size** – вычерчивание всех прямоугольников в одном размере, равным размеру самого большого прямоугольника в диаграмме дерева узлов, т.е. узла, у которого имеется самая длинная надпись ☐ **Border** - зона, в которой содержатся переключатели:

☐ **Include Kit** – задание заголовка в верхней части диаграммы узлов

☐ **Include title** – задание заголовка, содержащего номер узла, Сномер и номер страницы, в нижней части диаграммы

☐ **Connection Style** - зона, в которой содержатся переключатели:

☐ **Diagonal lines** – опция позволяющая создавать диагональные линии

☐ **Orthogonal lines** - опция позволяющая создавать горизонтальные и вертикальные линии для соединения узлов диаграммы

• **Back** – кнопка перехода к предыдущему диалоговому окну мастера

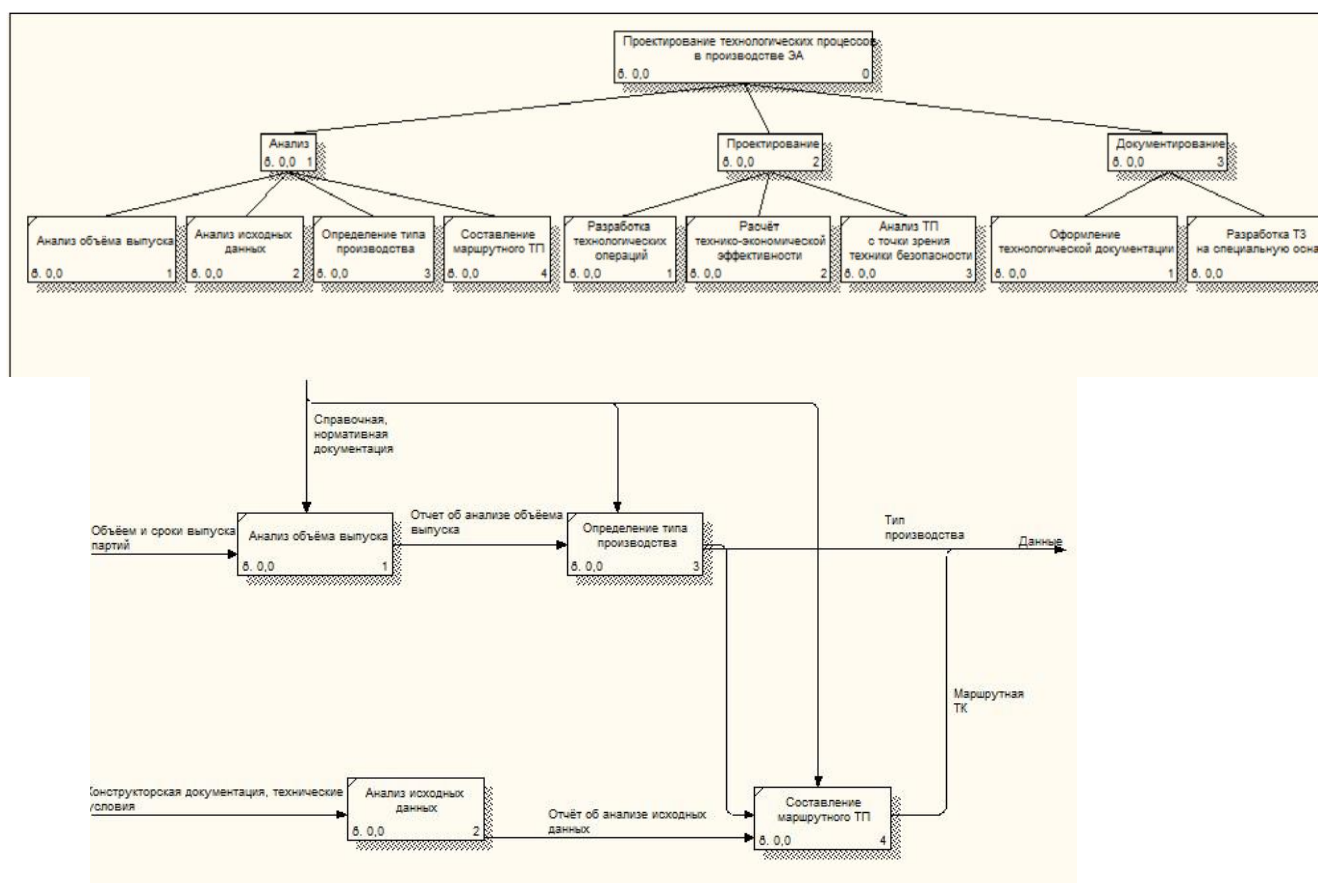
• **Next** - кнопка перехода к следующему диалоговому окну

• **Finish** – кнопка закрытия диалогового окна и добавление диаграммы узлов

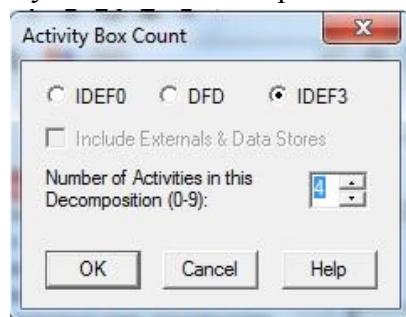
В результате настроек будет создана диаграмма дерева узлов:

Диаграмму дерева узлов можно модифицировать. Нижний уровень может быть отображен не в виде списка, а в виде прямоугольников, также, как и верхние уровни. Для модификации диаграммы правой кнопкой мыши щелкните по свободному месту, не занятому объектами, выберите меню **Node Tree Diagram Properties** и во вкладке **Style** диалога

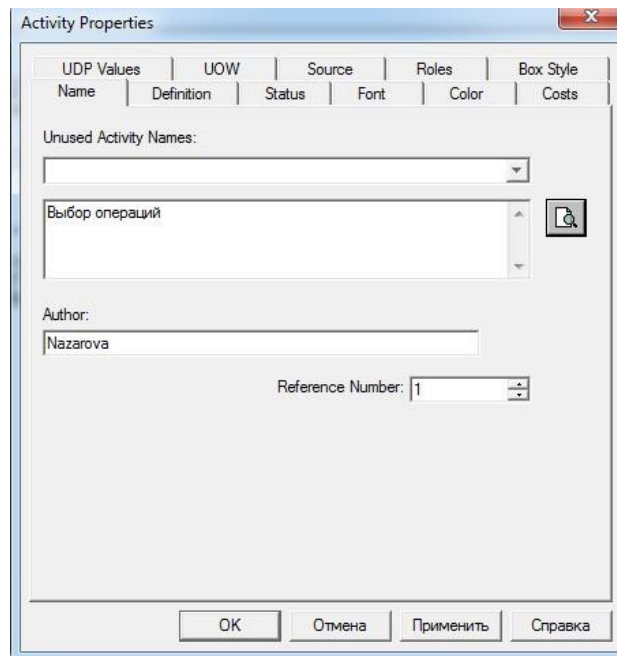
Node Tree Properties отключите опцию **Bullet last level**. Пример создания диаграммы потоков работ. Перейдите на диаграмму A1 и декомпозируйте одну из работ, например:



В диалоге **ActivityBoxCount** установите число работ и нотацию **IDEF3**.



Возникает диаграмма **IDEF3**, содержащая работы **Unit of Work (UOW)**, также называемыми единицами работы или работами. Правой кнопкой мыши щелкните по работе с номером 1, выберите в контекстном меню **Name** и внесите имя работы.



Activity Properties

UDP Values | UOW | Source | Roles | Box Style

Name | Definition | Status | Font | Color | Costs

Unused Activity Names:

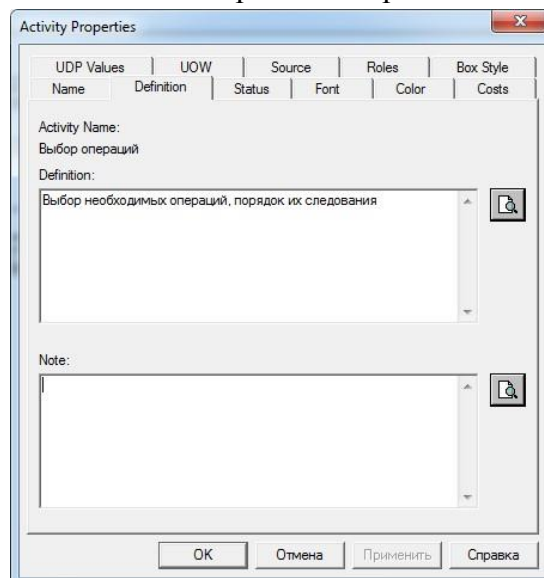
Выбор операций

Author: Nazarova

Reference Number: 1

OK Отмена Применить Справка

Затем во вкладке **Definition** внесите определение работы с номером 1



Activity Properties

UDP Values | UOW | Source | Roles | Box Style

Name | Definition | Status | Font | Color | Costs

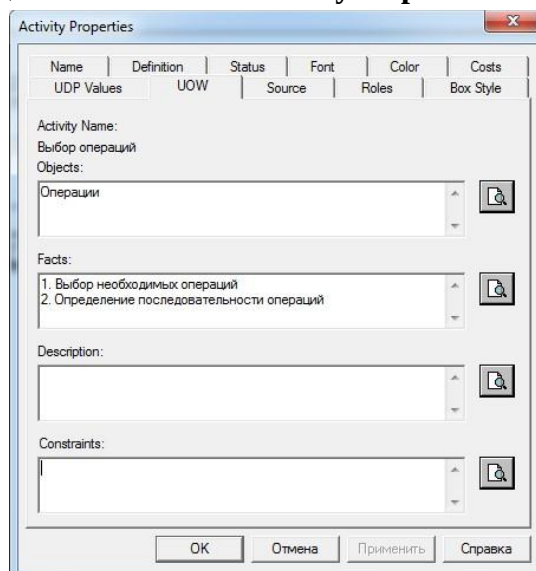
Activity Name: Выбор операций

Definition: Выбор необходимых операций, порядок их следования

Note:

OK Отмена Применить Справка

Во вкладке **UOW** диалогового окна **Activity Properties** внесите свойства работы 1.



Activity Properties

Name | Definition | Status | Font | Color | Costs

UDP Values | UOW | Source | Roles | Box Style

Activity Name: Выбор операций

Objects: Операции

Facts: 1. Выбор необходимых операций 2. Определение последовательности операций

Description:

Constraints:

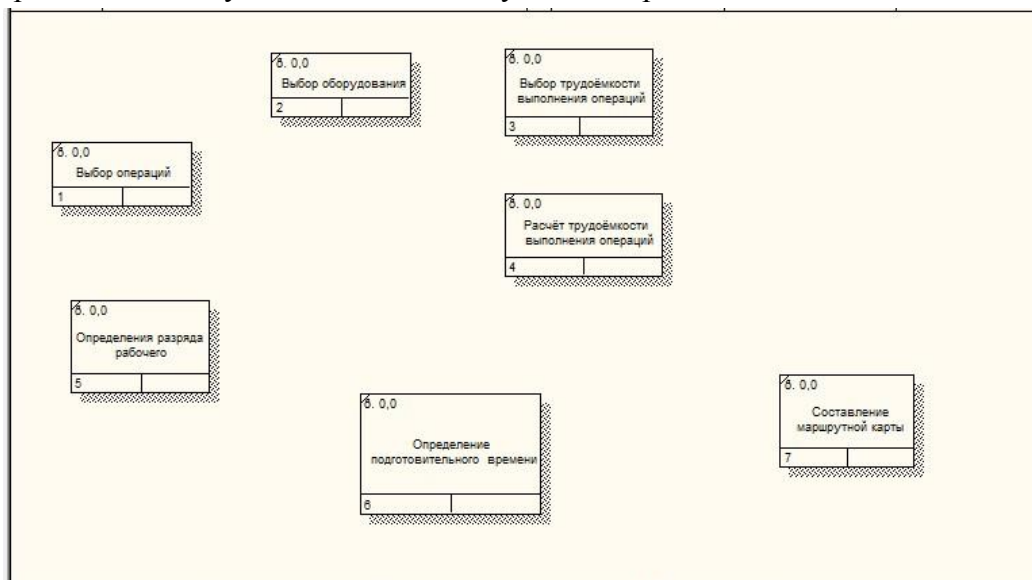
OK Отмена Применить Справка

Внесите в диаграмму ещё три работы (с помощью кнопки 

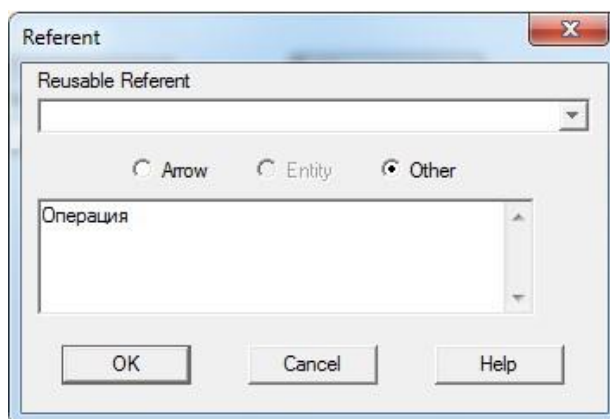
Activity Box Tool)

и присвойте имена работам с номерами 2....7

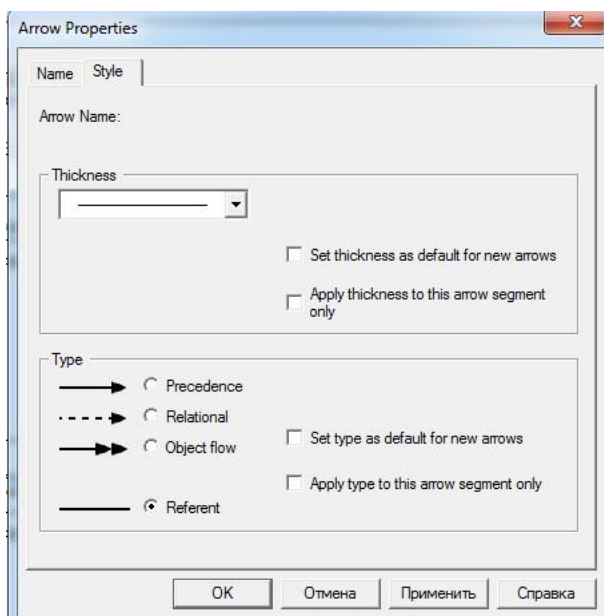
Диаграмма **IDEF3** будет выглядеть следующим образом



С помощью кнопки  **Referent Tool** палитры инструментов создайте объект ссылки. Внесите имя объекта внешней ссылки

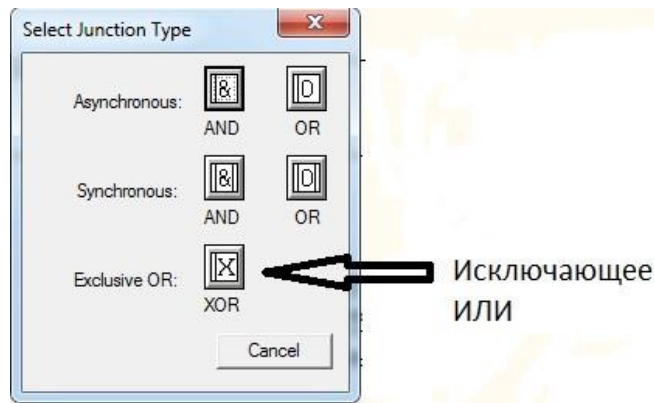


Свяжите стрелкой объект ссылки и соответствующую работу. Измените вид стрелки, связывающий объект ссылки работу, воспользовавшись диалоговым окном **ArrowProperties**.

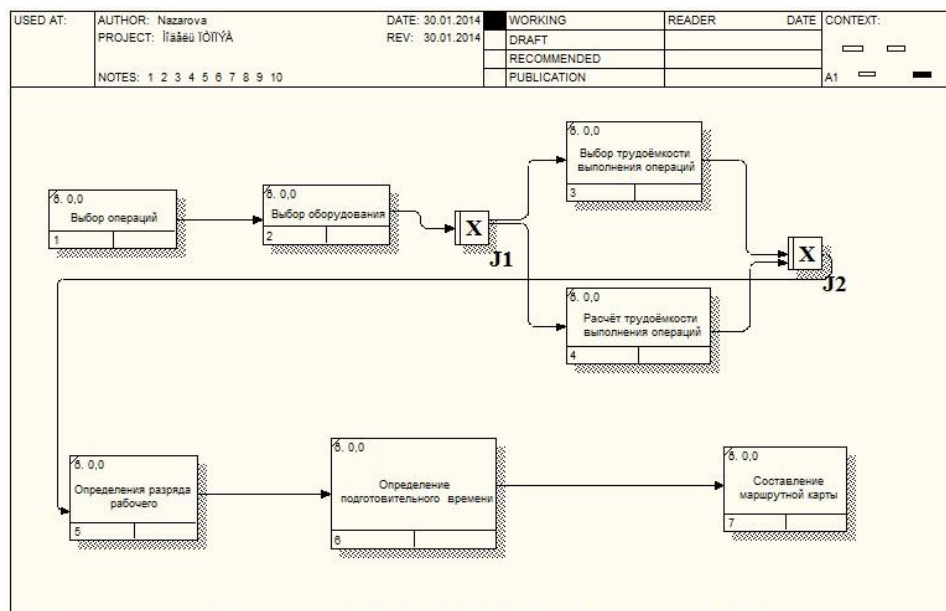


Свяжите стрелкой 1 и 2 работы.

С помощью кнопки  **Junction Tool** на палитре инструментов внесите перекрёсток типа «Исключающее ИЛИ»



Свяжите работы с перекрёстком. Так будет выглядеть завершённая диаграмма IDEF3:

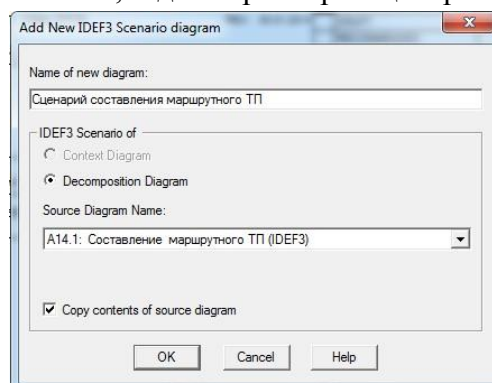


Практическая работа №13. Порядок создания сценария.

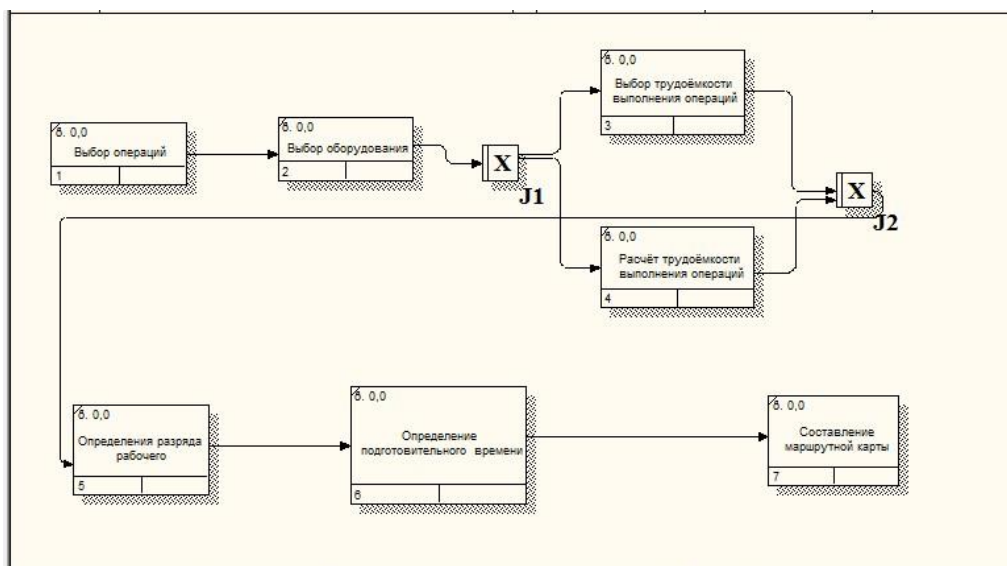
Цель: Изучить основные определения и понятия методологии IDEF3. Познакомиться с основными способами построения диаграммы дерева узлов. Научиться строить диаграммы потоков работ и диаграммы дерева узлов.

Порядок создания сценария

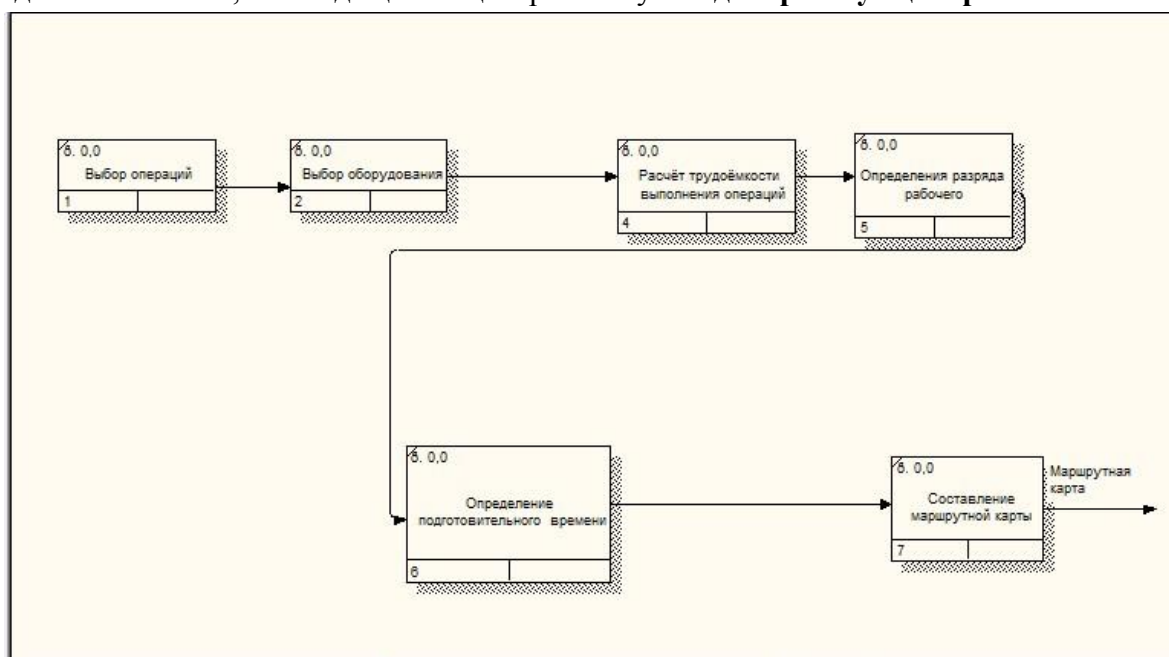
Выберите пункт главного меню **Diagram/AddIDEF3 Scenario**. Создайте диаграмму сценария на основе диаграммы IDEF3, задав параметры сценария.



Проект сценария будет выглядеть так:



Удалив элементы, не входящие в сценарий получим **диаграмму сценария**



Контрольные вопросы:

1. Что такое диаграммы дерева узлов? Для чего они используются в моделировании бизнес-процессов?
2. Что такое диаграмма потоков работ? Для чего они используются в моделировании бизнес-процессов?
3. Назовите основные объекты диаграммы дерева узлов?
4. Назовите основные объекты диаграммы потоков работ?
5. Перечислите типы перекрёстков, используемые в диаграмме потоков работ?
6. Что такое объект ссылки? Типы стилей объектов?

Практическая работа №14.

Создание диаграммы потоков данных (DFD) с помощью инструментальной среды Ramus educational.

Цель: Рассмотреть основные понятия и определения методологии DFD. Научиться строить диаграммы потоков данных в инструментальной среде Ramuseducational.

В результате освоения учебной дисциплины обучающийся должен **уметь**:

□ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;

В результате освоения учебной дисциплины обучающийся должен **знать**:

□ модели жизненного цикла информационной системы, методы проектирования информационной системы;

□ технологии проектирования информационной системы, оценку и управление качеством информационной системы;

□ организацию труда при разработке информационной системы;

□ оценку необходимых ресурсов для реализации проекта.

Теоретическая часть

Диаграммы потоков данных (Data Flow Diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. **DFD описывает:**

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

В Brwin для построения диаграмм потоков данных используется нотация Гейна-Сарсона.

Для того чтобы дополнить модель IDEF0 диаграммой DFD, нужно в процессе декомпозиции в диалоге Activity Box Count «кликнуть» по радиокнопке DFD. В панели инструментов на новой диаграмме DFD появляются новые кнопки:

- добавить в диаграмму внешнюю ссылку (External Reference). Внешняя ссылка является источником или приемником данных извне модели;
- добавить в диаграмму хранилище данных (Data store). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах;
- ссылка на другую страницу. В отличие от IDEF0 инструмент offpage reference позволяет направить стрелку на любую диаграмму (а не только на верхний уровень).

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более похожими на физические характеристики системы – движение объектов (data flow), хранение объектов (data stores), поставка и распространение объектов (external entities).

В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов. Контекстная диаграмма часто включает работы и внешние ссылки. Работы обычно именуются по названию системы, например **«Система обработки информации»**. Включение внешних ссылок в

контекстную диаграмму не отменяет требования методологии четко определить цель, область и единую точку зрения на моделируемую систему.

Работы. В DFD работы представляют собой функции системы, преобразующие входы в выходы. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ IDEF0 и IDEF3. Так же, как работы IDEF3, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0.

Внешние сущности. Внешние сущности изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

Стрелки (Потоки данных). Стрелки описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника работы. В DFD также применяются двунаправленные стрелки для описания диалогов типа «команда-ответ» между работами, между работой и внешней сущностью и между внешними сущностями (рис. 2).

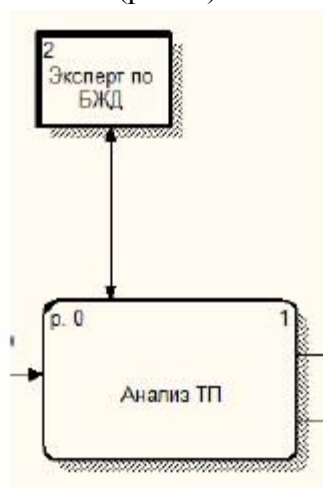


Рис. 1. Внешняя сущность

Хранилище данных. В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое.

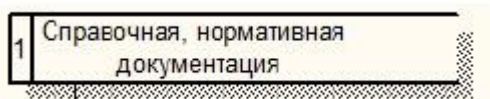


Рис. 2. Хранилище данных

В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов.

Слияние и разветвление стрелок. В DFD стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Построение диаграмм DFD. Диаграммы DFD могут быть построены с использованием традиционного структурного анализа, подобно тому, как строятся диаграммы IDEF0. Сначала строится физическая модель, отображающая текущее состояние дел. Затем эта модель преобразуется в логическую модель, которая отображает требования к существующей системе. После этого строится модель, отображающая

требования к будущей системе. И наконец, строится физическая модель, на основе которой должна быть построена новая система.

Альтернативным подходом является подход, популярный при создании программного обеспечения, называемый **событийным разделением** (event partitioning), в котором различные диаграммы DFD выстраивают модель системы. **Во-первых**, логическая модель строится как совокупность работ и документирования того, что они (эти работы) должны делать.

Затем модель окружения (environment model) описывает систему как объект, взаимодействующий с событиями из внешних сущностей. **Модель окружения** обычно содержит описание цели системы, одну контекстную диаграмму и список событий. Контекстная диаграмма содержит один прямоугольник работы, изображающий систему в целом, и внешние сущности, с которыми система взаимодействует.

Наконец, модель поведения (behavior model) показывает, как система обрабатывает события. Эта модель состоит из одной диаграммы, в которой каждый прямоугольник изображает каждое событие из модели окружения. Хранилища могут быть добавлены для моделирования данных, которые необходимо запоминать между событиями. Потоки добавляются для связи с другими элементами, и диаграмма проверяется с точки зрения соответствия модели окружения.

Полученные диаграммы могут быть преобразованы с целью более наглядного представления системы, в частности работы на диаграммах могут быть декомпозированы.

Нумерация объектов. В DFD номер каждой работы может включать префикс, номер родительской работы (A) и номер объекта. **Номер объекта** – это уникальный номер работы на диаграмме. Например, работа может иметь номер A.12.4. Уникальный номер имеют хранилища данных и внешние сущности независимо от их расположения на диаграмме. Каждое хранилище данных имеет префикс D и уникальный номер, например D5. Каждая внешняя сущность имеет префикс E и уникальный номер, например E5.

Практическая работа №15.

Работа с диаграммой потоков данных (DFD) с помощью инструментальной среды Ramus educational.

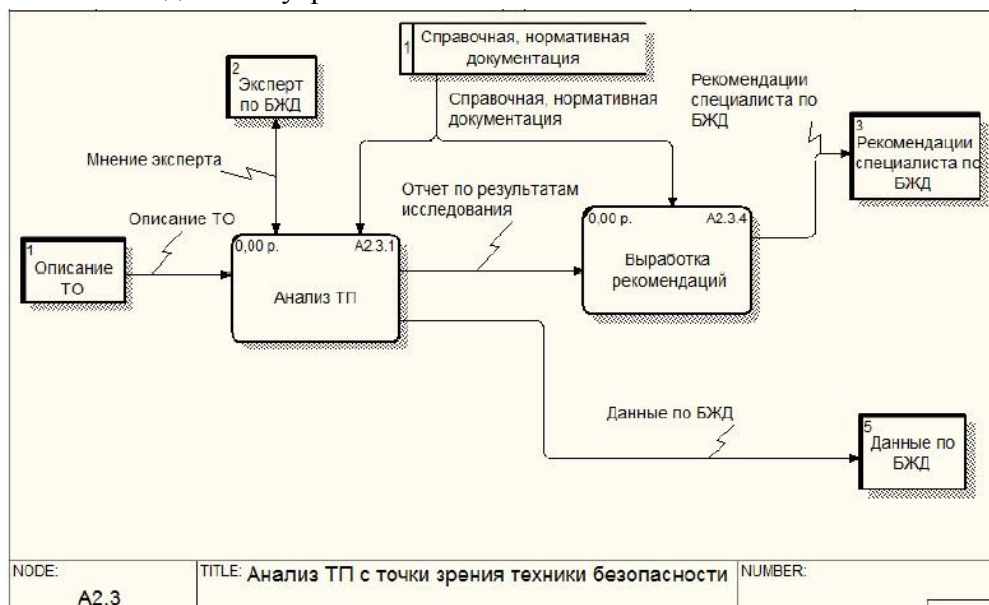
Цель: Рассмотреть основные понятия и определения методологии DFD. Научиться строить диаграммы потоков данных в инструментальной среде Ramuseducational.

Практическая часть

Задание: На основе, построенных в предыдущих практических работах контекстной диаграммы и диаграмм декомпозиций, создать диаграмму потоков данных.

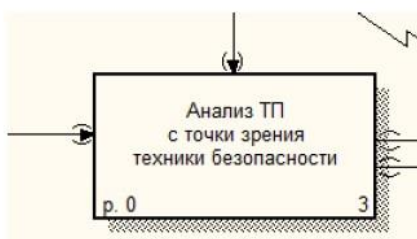
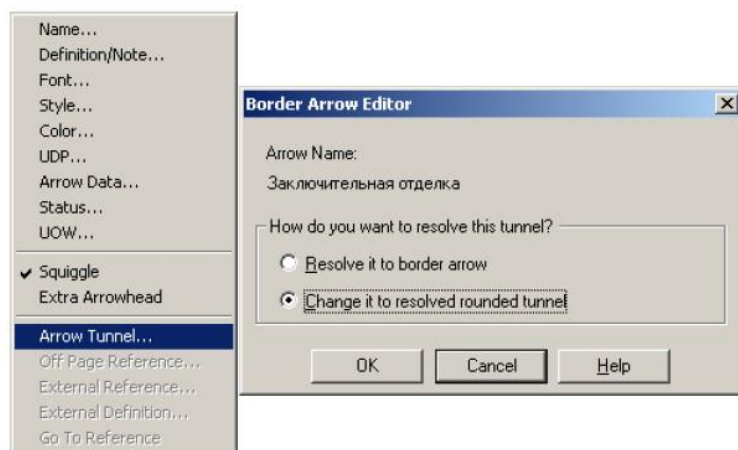
Пример создания диаграммы потоков данных. В процессе декомпозиции согласно правилам DFD необходимо преобразовать граничные стрелки во внутренние, начинающиеся и заканчивающиеся на внешних ссылках.

1. Декомпозируйте одну из работ на диаграмме A2.
2. В диалоге Activity Box Count выберите количество работ 2 и нотацию DFD.
3. Щелкните по ОК и внесите в новую диаграмму DFD A23 имена работ.
4. Внесите хранилище  **Data store Tool**.
5. Внесите не менее трёх внешних ссылок  **External Reference Tool**.
6. Создайте внутренние ссылки.



Обратите внимание, что стрелка «*Мнение эксперта*» двунаправленная. Для того чтобы сделать стрелку двунаправленной, щелкните правой кнопкой по стрелке, выберите в контекстном меню пункт **Style** и во вкладке **Style** выберите опцию **Bidirectional**.

На родительской диаграмме A2 туннелируйте стрелки, подходящие и исходящие из работы «*Анализ ТП с точки зрения техники безопасности*». Для этого щелкните правой клавишей мыши по концу стрелки, прилегающему к данной работе, выберите **Arrow Tunnel...** и в появившемся диалоговом окне выберите **Change it to resolved rounded tunnel**.



Практическая работа №16. Использование Off-Page Reference.

Цель: Рассмотреть основные понятия и определения методологии DFD. Научиться строить диаграммы потоков данных в инструментальной среде Ramuseducational.

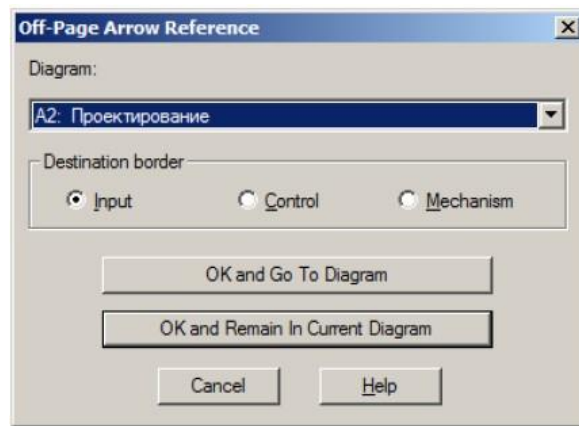
Использование Off-Page Reference на диаграмме DFD

Некоторые стрелки с диаграмм IDEF0 и DFD (не только с родительских) могут показываться на диаграмме DFD. Для отображения таких стрелок используется инструмент **Off-Page Reference**.

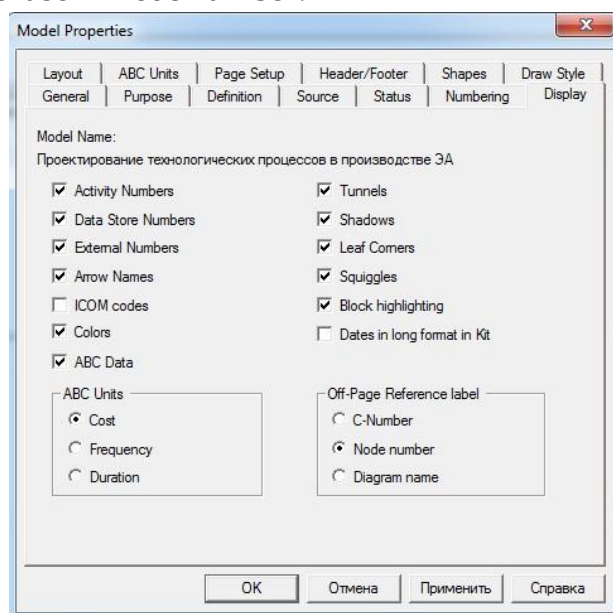
В случае, если в результате выработки рекомендаций выявляется, что технологическая операция устаревшая и существует более новая и совершенная, то требуется обновить технологическую операцию. Для этого необходимо использовать инструмент **Off-Page Reference**. На диаграмме A23 создайте новую граничную стрелку.



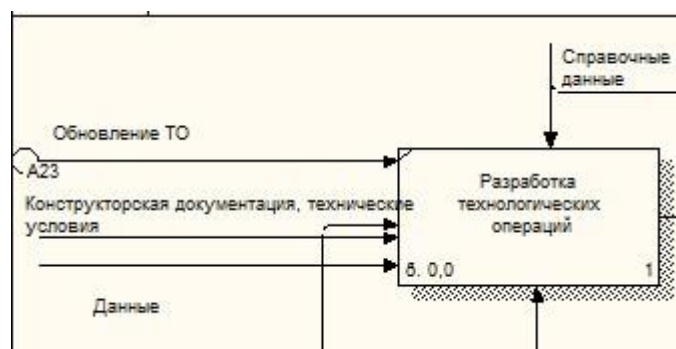
Правой кнопкой щелкните по наконечнику стрелки и выберите в меню **Off-Page Reference**. В появившемся диалоге **Off-Page Arrow Reference** выберите в качестве диаграммы A2.



Перейдите в меню **Model/Model Properties**, далее – во вкладку **Display**. Установите опцию Off-Page Reference label – **Node number**.



Перейдите на диаграмму **A2** и направьте новую стрелку на вход первой работы.



Контрольные вопросы:

1. Что описывает диаграмма DFD?
2. Какая нотация используется в Ramuseducational для построения диаграммы DFD?
3. Из каких элементов состоит диаграмма DFD?

4. Что называется внешней сущностью?
5. Что описывают хранилища?
6. Что представляют собой стрелки DFD?
7. Опишите механизм дополнения диаграммы IDEF0 диаграммой DFD?

Практическая работа №17. Построение сетевых графиков.

Цель: Изучить основные понятия сетевого моделирования и теории графов. Научиться находить оптимальное решение той или производственной или коммерческой ситуации. Уметь строить графики заданного комплекса работ, рассчитывать временные характеристики сетевого графика при нормальном режиме работ; находить критический путь, полные резервы времени, временные характеристики сетевого графика при срочном режиме работ, критический путь, полные резервы времени, стоимость работ.

В результате освоения учебной дисциплины обучающийся должен **уметь**:

☐ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;

☐ использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения.

В результате освоения учебной дисциплины обучающийся должен **знать**:

☐ реинжиниринг бизнес-процессов;

☐ требования к проектируемой системе;

☐ модели жизненного цикла информационной системы, методы проектирования информационной системы;

☐ технологии проектирования информационной системы, оценку и управление качеством информационной системы;

☐ организацию труда при разработке информационной системы;

☐ оценку необходимых ресурсов для реализации проекта.

Теоретическая часть

Поиск более эффективных способов планирования сложных процессов привели к необходимости использования моделей сетевого планирования и управления (СПУ). СПУ основано на моделировании процесса с помощью сетевого графика (сетевой модели).

Сетевая модель представляет план выполнения некоторого комплекса работ. Главными элементами сетевого графика является событие и работа. **Событие** – это завершение, какого либо процесса, отражающий отдельный этап выполнения проекта. На сетевом графике событие изображается кружком.

Работа – процесс, связанный с затратами времени и ресурсов и приводящий к получению определённых результатов. Работами считают также процессы, не требующие расходов ресурса, а только времени. **Действительная работа** — протяжённый во времени процесс, требующий затрат ресурсов (например, сборка изделия, испытание прибора и т. п.).

Каждая действительная работа должна быть конкретной, чётко описанной и иметь ответственного исполнителя.

Зависимость, или фиктивная работа — логическая связь между двумя или несколькими работами (событиями), не требующими затрат труда, материальных ресурсов или времени. Она указывает, что возможность одной работы непосредственно зависит от результатов другой. Естественно, что продолжительность фиктивной работы принимается равной нулю.

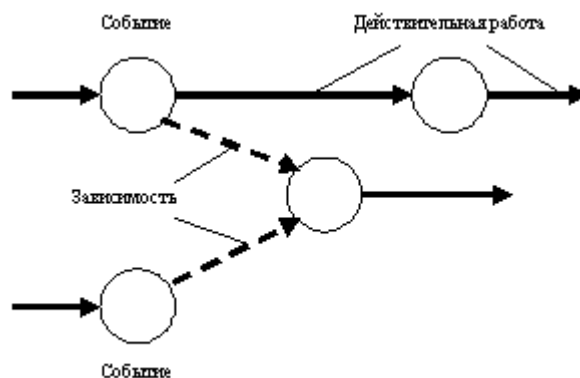


Рис. 1 Основные элементы сетевой модели

При составлении сетевых графиков (моделей) используют условные обозначения. События на сетевом графике изображаются кружками (вершинами графа), а работы — стрелками (ориентированными дугами):

○ — событие,

→ — работа (процесс),

-----> — фиктивная работа — применяется для упрощения

сетевых графиков (продолжительность всегда равна 0).

Среди событий сетевой модели выделяют **исходное** и **завершающее события**. **Исходное событие** не имеет предшествующих работ и событий, относящихся к представленному в модели комплексу работ. **Завершающее событие** не имеет последующих работ и событий.

Существует и иной принцип построения сетей — без событий. В такой сети вершины графа означают определённые работы, а стрелки — зависимости между работами, определяющие порядок их выполнения. Сетевой график **«работы–связи»** в отличие от графика **«события–работы»** обладает известными преимуществами: не содержит фиктивных работ, имеет более простую технику построения и перестройки, включает только хорошо знакомое исполнителям понятие работы без менее привычного понятия события.

Если в сетевой модели нет числовых оценок, то такая сеть называется **структурной**. Однако на практике чаще всего используют сети, в которых заданы оценки продолжительности работ, а также оценки других параметров, например трудоёмкости, стоимости и т. п.

Известны два метода сетевого планирования: **СРМ** — метод критического пути при фиксированном числе работ для составления расписания, и **PERT** — метод оценки и пересмотра программ.

Правила построения классических сетевых графиков

При построении сетевого графика: рекомендуется направлять стрелки слева направо и изображать их по возможности горизонтальными линиями без лишних пересечений.

Для правильного отображения взаимосвязи между работами сетевого графика при его построении необходимо соблюдать ряд правил.

Первое правило. Если работы А, Б и В выполняются последовательно, то на сетевом графике изображаются по горизонтали одна за другой (рисунок 2).

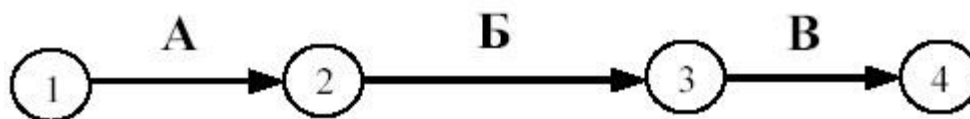


Рис. 2 Первое правило

Второе правило. Если результат работы А необходим для выполнения работ Б и В, то на сетевом графике это отображается следующим образом (рисунок 3).

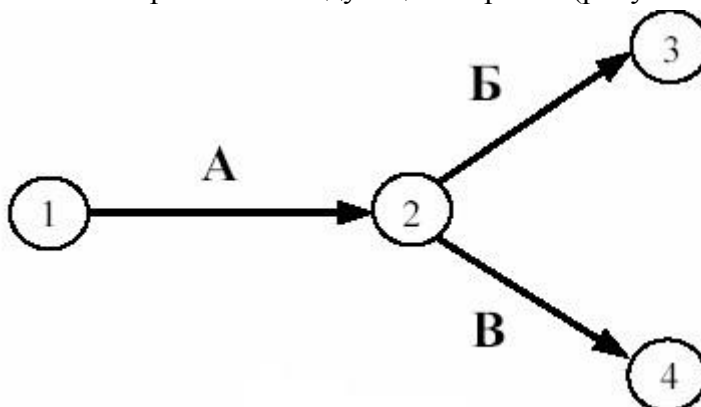


Рис.3 Второе правило

Третье правило. Если результат работ Г и Д необходим для выполнения работы Е, то на сетевом графике это изображается так (рисунок 4).

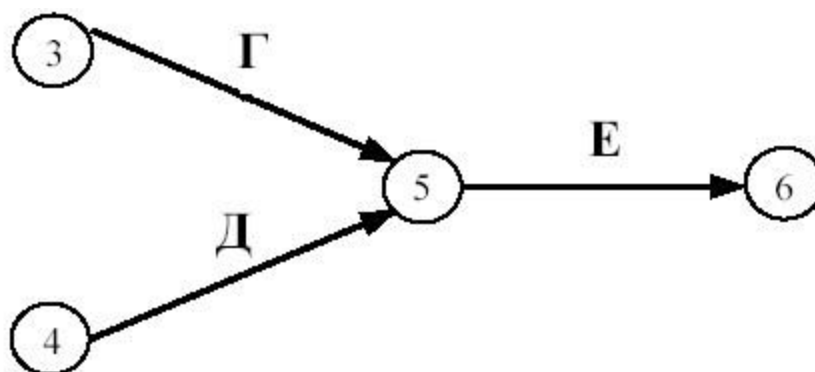
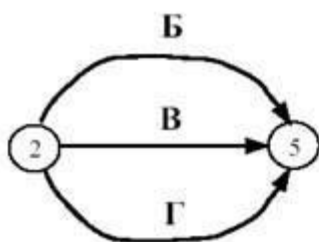
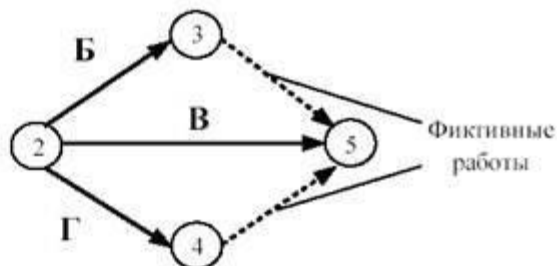


Рис. 4 Третье правило

Четвертое правило. Работы сетевого графика не должны иметь одинакового кода. Если работы Б, В, Г выходят из одного события и выполнение необходимо для свершения одного и того же события, то вводятся дополнительные фиктивные работы (рисунок 5).



а – неправильно



б – правильно

Рис. 5 Четвертое правило

Пятое правило. Если работы Б, В и Г начинаются после частичного выполнения работы А, то работа А разбивается на части А1, А2 ... Аi и т.д., при этом каждая работа А в сетевом графике считается самостоятельной работой (рисунок 6).

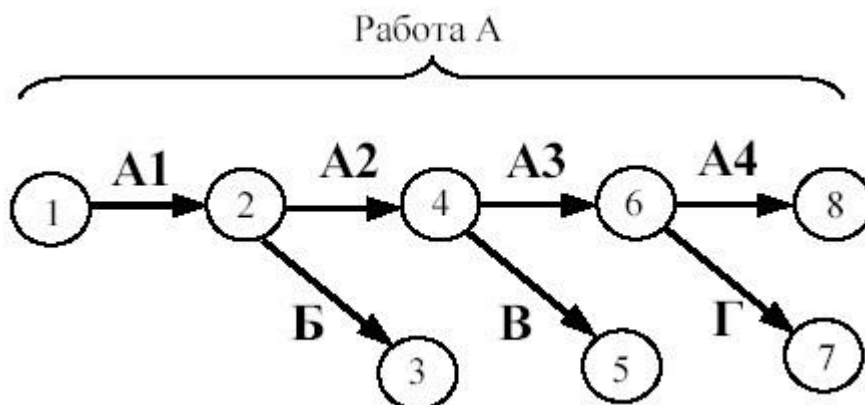


Рис. 6 Пятое правило

Шестое правило. Если для начала работы Ж необходимо выполнение работ В и Г, а для начала работы Д выполнение работы Г, то в сетевой график вводится дополнительная фиктивная работа (рисунок 7).

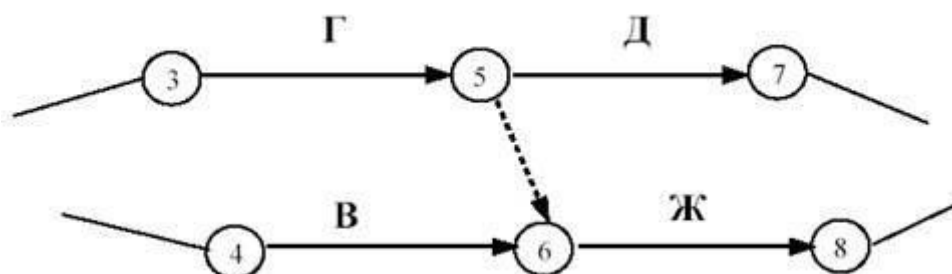


Рис. 7 Шестое правило

Седьмое правило. Если после окончания работы А можно начать работу Б, а после окончания работы В работу Г, а работа Д может быть начата только после окончания работ А и В, то на сетевом графике это изображается с помощью двух дополнительных фиктивных работ (рисунок 8).

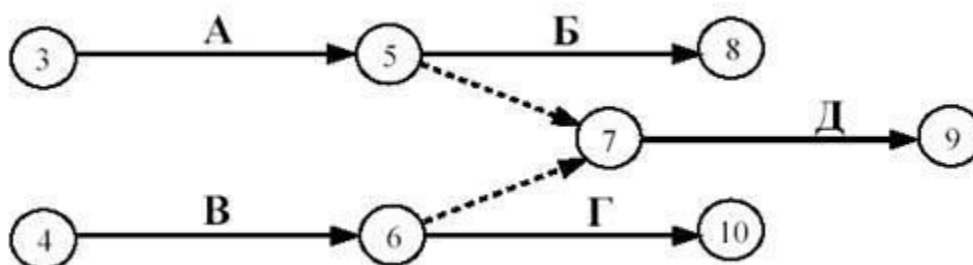


Рис. 8 Седьмое правило

Восьмое правило. В сетевом графике не должно быть замкнутых контуров (циклов), т.е. цепочек работ, возвращающихся к тому событию, из которого они вышли. На рисунке 9 замкнутый контур (цикл) образовался из событий 3, 4, 2, 3. Наличие цикла в сети свидетельствует об ошибке в исходных данных или в неправильном изображении взаимосвязи работ.

Такая ситуация чаще возникает в больших и сложных сетях, которые разрабатываются несколькими исполнителями. При обнаружении подобной ошибки сетевой графика, после выяснения ее причины, необходимо исправить (рисунок 9).

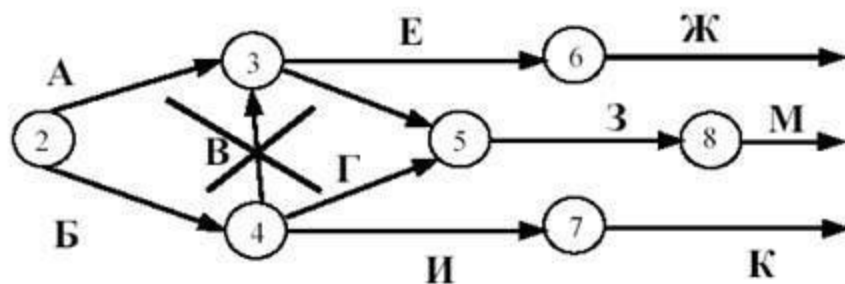
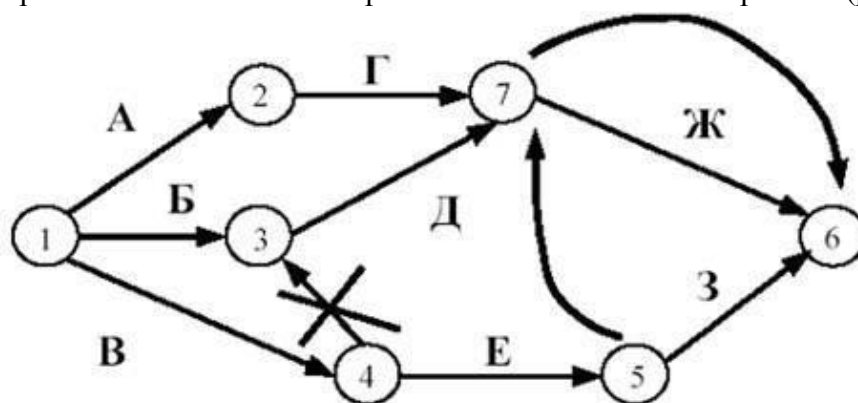
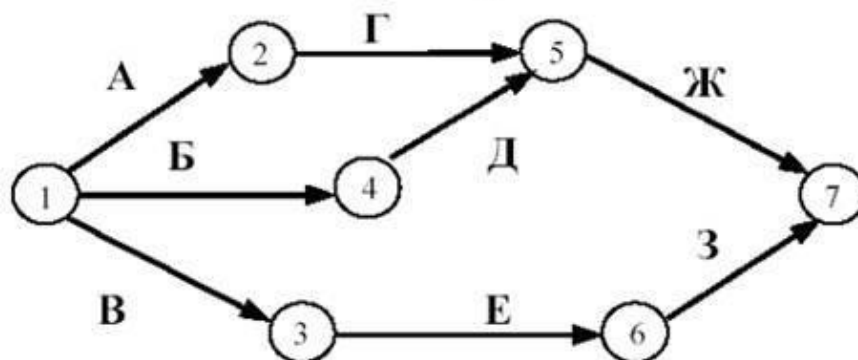


Рис. 9 Восьмое правило

Девятое правило. События следует кодировать так, чтобы номер начального события данной работы был меньше номера конечного события этой работы (рисунок 10).



неправильно



правильно

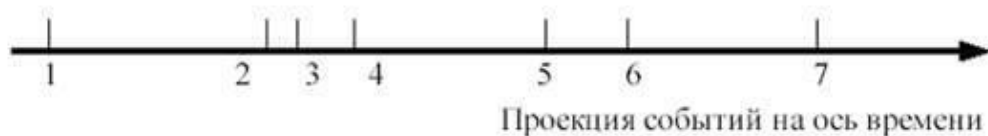


Рис.10 Девятое правило

Десятое правило. В одноцелевом сетевом графике не должно быть "тупиков", т.е. таких событий, из которых не выходит ни одной работы. Если в сети, кроме завершающего, появилось еще одно событие, из которого не выходит ни одной работы – это означает либо ошибку при построении сетевого графика, либо планирование ненужной работы Б, результат которой никого не интересует (рисунок 11).

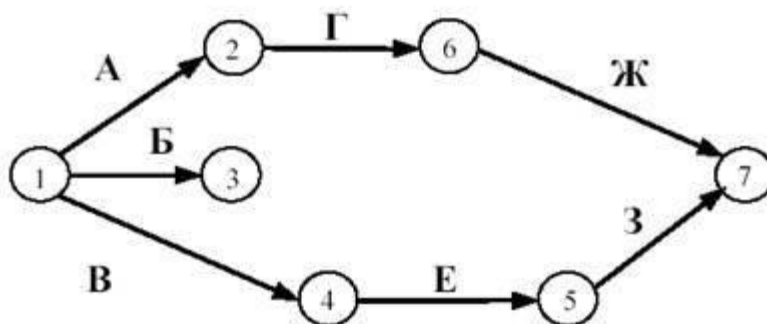


Рис. 11 Десятое правило

Одиннадцатое правило. В сетевом графике не должно быть "хвостов", т.е. событий, в которые не входит ни одной работы, если эти события не являются исходными для данного сетевого графика. Если это правило нарушено, и в сети, кроме исходного, появилось еще одно событие, в которое не входит ни одной работы – это означает либо ошибку при составлении сетевого графика, либо отсутствие работы, результат которой необходим для начала работы (Г).

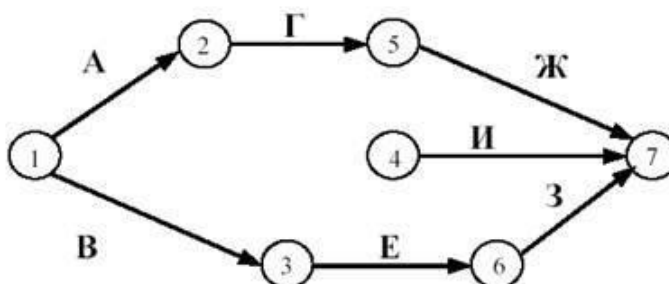


Рис. 12 Одиннадцатое правило

Двенадцатое правило. При укрупнении сетевых графиков группа работ может изображаться как одна работа, если в этой группе имеется одно конечное событие и если эти работы выполняются одним исполнителем при наличии в группе входных и выходных работ.

Продолжительность укрупненной работы равна продолжительности наибольшего пути от начального до конечного событий этой группы работ.

Практическая работа №18.

Работа с сетевыми графиками, понятие пути.

Цель: Изучить основные понятия сетевого моделирования и теории графов. Научиться находить оптимальное решение той или производственной или коммерческой ситуации. Уметь строить графики заданного комплекса работ, рассчитывать временные характеристики сетевого графика при нормальном режиме работ; находить критический путь, полные резервы времени, временные характеристики сетевого графика при срочном режиме работ, критический путь, полные резервы времени, стоимость работ.

Понятие о пути

Одно из важнейших понятий сетевого графика — понятие пути. **Путь — любая последовательность работ, в которой конечное событие каждой работы совпадает с начальным событием следующей за ней работы.** Среди различных путей сетевого графика наибольший интерес представляет **полный путь** — любой путь, начало которого совпадает с исходным событием сети, а конец — с завершающим.

Наиболее продолжительный полный путь в сетевом графике называется **критическим**. Критическими называются также работы и события, находящиеся на этом пути.

Условия критичности пути

- **необходимое условие:** нулевые резервы событий, лежащих на критическом пути $R(i) \equiv 0$;
- **достаточное условие:** нулевые полные резервы работ, лежащих на критическом пути $R_n(i, j) \equiv 0$. $R_n(i, j) \equiv T_n(j) \equiv T_p(i) \equiv t(i, j)$ — показывает максимальное время, на которое можно увеличить длительность работы (i, j) или отсрочить ее начало, чтобы не нарушился срок завершения проекта в целом.

На стадии управления и контроля над ходом выполнения программы основное внимание уделяется работам, находящимся на критическом пути или в силу отставания попавшим на критический путь. Для сокращения продолжительности проекта необходимо в первую очередь сокращать продолжительность работ, лежащих на критическом пути.

Временные параметры сетевых графиков

Ранний (или ожидаемый) срок свершения события определяется продолжительностью максимального пути, предшествующего этому событию.

Задержка свершения события по отношению к своему раннему сроку не отразится на сроке свершения завершающего события (а значит, и на сроке выполнения комплекса работ) то тех пор, пока сумма срока свершения этого события и продолжительности (длины) максимального из последующих за ним путей не превысит длины критического пути.

Поэтому **поздний (или предельный) срок свершения события** равен разности максимального времени наступления последующего за работой события и времени работы до этого (будущего) события.

Резерв времени события определяется как разность между поздним и ранним сроками его свершения.

Резерв времени события показывает, на какой допустимый период времени можно задержать наступление этого события, не вызывая при этом увеличения срока выполнения комплекса работ.

Критические события резервов времени не имеют, так как любая задержка в свершении события, лежащего на критическом пути, вызовет такую же задержку в свершении завершающего события.

Из этого следует, что для того, чтобы определить длину и топологию критического пути, вовсе не обязательно перебирать все полные пути сетевого графика и определять их длины. Определив ранний срок наступления завершающего события сети, мы тем самым определяем длину критического пути, а, выявив события с нулевыми резервами времени, определяем его топологию.

Если сетевой график имеет единственный критический путь, то этот путь проходит через все критические события, то есть события с нулевыми резервами времени. Если критических путей несколько, то выявление их с помощью критических событий может быть затруднено, так как через часть критических событий могут проходить как критические, так и некритические пути. В этом случае для определения критических путей рекомендуется использовать критические работы.

Отдельная работа может начаться (и закончиться) в ранние, поздние или другие промежуточные сроки. В дальнейшем при оптимизации графика, возможно любое размещение работы в заданном интервале, называемом **продолжительностью работы**.

Очевидно, что **ранний срок начала работы** совпадает с **ранним сроком наступления предшествующего события**.

Ранний срок окончания работы совпадает с **ранним сроком свершения последующего события**.

Поздний срок начала работы совпадает с **поздним сроком наступления предшествующего события**.

Поздний срок окончания работы совпадает с **поздним сроком наступления последующего события**.

Таким образом, в рамках сетевой модели моменты начала и окончания работы тесно связаны с соседними событиями соответствующими ограничениями.

Если путь не критический, то он имеет **резерв времени, определяемый как разность между длиной критического пути и рассматриваемого**. Он показывает, на сколько в сумме могут быть увеличены продолжительности всех работ, принадлежащих этому пути. Отсюда можно сделать вывод, что любая из работ пути на его участке, не совпадающем с критическим путём (замкнутым между двумя событиями критического пути), обладает резервом времени. *Среди резервов времени работ выделяют четыре разновидности:*

Полный резерв времени работы показывает, на сколько можно увеличить время выполнения данной работы при условии, что срок выполнения комплекса работ не изменится.

Полный резерв времени работы равен резерву максимального из путей, проходящего через данную работу. Этим резервом можно располагать при выполнении данной работы, если её начальное событие свершится в самый ранний срок, и можно допустить свершение конечного события в его самый поздний срок. Остальные резервы времени работы являются частями её полного резерва.

Частный резерв времени первого вида есть часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом позднего срока её начального события. Этим резервом можно располагать при выполнении данной работы в предположении, что её начальное и конечное события свершаются в свои самые поздние сроки.

Частный резерв времени второго вида, или свободный резерв времени работы представляет часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом раннего срока её конечного события. Этим резервом можно располагать при выполнении данной работы в предположении, что её начальное и конечное события свершатся в свои самые ранние сроки.

Независимый резерв времени работы — часть полного резерва времени, получаемая для случая, когда все предшествующие работы заканчиваются в поздние сроки, а все последующие работы начинаются в ранние сроки.

Фактически независимый резерв имеют лишь те работы, которые не лежат на максимальных путях, проходящих через их начальные и конечные события.

Работы, лежащие на критическом пути, так же как и критические события, резервов времени не имеют.

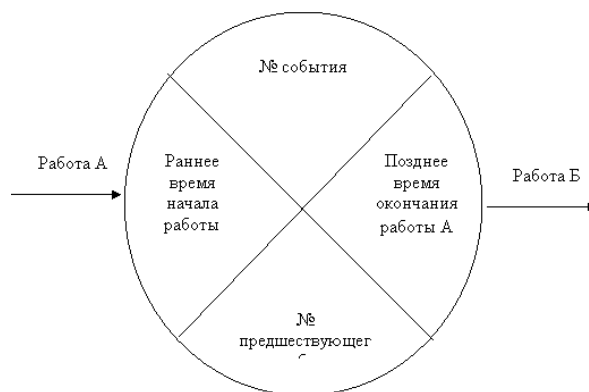


Рис. 13 Расчёт секторным методом

Следует отметить, что в случае достаточно простых сетевых графиков кроме табличного метода расчета параметров сетевых графиков, может быть применено секторное представление временных параметров, то есть расчет параметров может быть произведен на самом графике. Каждое событие для этого делится на четыре сектора. В левом секторе события записывают **раннее начало работы**, в правом — **позднее окончание**, в верхнем — **номер данного события**, в нижнем — **номер предшествующего события**, из которого к данному событию идёт путь максимальной продолжительности. Имеет место, когда в нижнем секторе ставят номер события и верхний сектор не заполняют. Определённые резервы времени записывают под стрелкой в виде дроби: в числителе общий резерв, а в знаменателе частный резерв.

Практическая работа №19.

Свойства сетевых графиков.

Цель: Изучить основные понятия сетевого моделирования и теории графов. Научиться находить оптимальное решение той или производственной или коммерческой ситуации. Уметь строить графики заданного комплекса работ, рассчитывать временные характеристики сетевого графика при нормальном режиме работ; находить критический путь, полные резервы времени, временные характеристики сетевого графика при срочном режиме работ, критический путь, полные резервы времени, стоимость работ.

Практическая часть

Основным временным параметром сетевого графика является продолжительность критического пути. Расчет критического пути состоит из двух этапов. Можно выделить:

1. **Прямой проход.** Вычисление начинают с исходного события и продолжают до тех пор, пока не будет достигнуто завершающее событие. Для каждого события вычисляется одно число, соответствующее раннему сроку наступления этого события;
2. **Обратный проход.** Вычисления начинают с завершающего события и продолжают, пока не будет достигнуто исходное событие. Для каждого события вычисляется поздний срок его наступления.

Рассмотрим прямой проход. Введем обозначения:

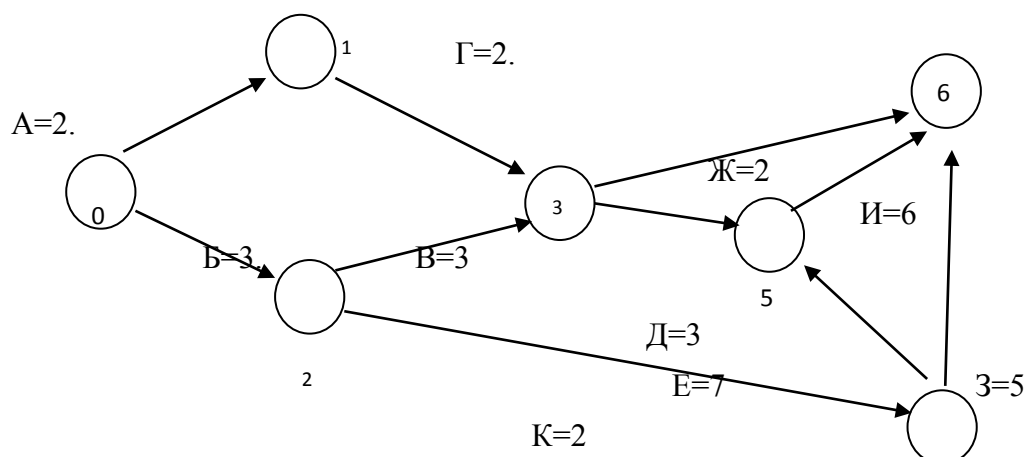
- $t_i^{рн}$ - ранний срок начала всех работ, следующих за событием i ;
- $t_j^{рн}$ - ранний срок начала всех работ, предшествующих событию j ; -
- t_{ij} продолжительность работы (i,j) ;
- $t_0^{рн} = 0$;
- $t_{ij}^{ро}$ - ранний срок окончания работы;
- $t_i^{по}$ - поздний срок окончания всех работ, предшествующих событию i ;
- $t_{ij}^{пн}$ - поздний срок начала работ; n – завершающее событие сети;
- $t_j^{рн} = \max_i (t_i^{рн} + t_{ij})$ для всех (i,j) .

Пример 1.

Для заданной сетевой модели некоторого комплекса работ определить время, критический путь и оптимальный путь, ранние сроки начала и окончания работ, поздние сроки начала и окончания работ, резервы времени.

Р абота	Описание работы	Непосредств енно предшествующие работы	Продолжител ьность работы (недели)
А , Б	Разработка технической документации (ТД) на прибор и его электронную часть.	-	А=2 Б=3
В , Г	Разработка технологической документации на электронную часть прибора и прибор.	А, Б	В=3 Г=2
Д	Передача ТД на прибор.	Г	Д=3
Е	Изготовление приборов.	В	Е=7
Ж	Изготовление электронной части прибора.	Г, Д	Ж=2
З , К	Разработка ТД на эксплуатацию прибора и электронную часть.	В, Г	З=5 К=2
И	Сборка и испытание прибора	Е, Ж	И=6

На основе данных таблицы построим сетевой график создания прибора с учетом изложенных рекомендаций.



Для данного графа выпишем частичные и полные пути, определим критический путь и оптимальный.

Частичные пути:

$$L_1 = (0, 1)$$

$$L_2 = (0, 1, 3)$$

$$L_3 = (0, 1, 3, 5)$$

$$L_4 = (0, 2)$$

$$L_5 = (0, 2, 3)$$

$$L_6 = (0, 2, 3, 5)$$

$$L_7 = (0, 2, 4)$$

$$L_8 = (0, 2, 4, 5)$$

Полные пути:

$$L_1 = (0, 1, 3, 6)$$

$$L_2 = (0, 1, 3, 5, 6)$$

$$L_3 = (0, 2, 3, 6)$$

$$L_4 = (0, 2, 3, 5, 6)$$

$$L_5 = (0, 2, 4, 5, 6)$$

$$L_6 = (0, 2, 4, 6)$$

Определим ранние сроки начала всех работ из предыдущего примера. Для исходного события ранний срок начала работ равен позднему сроку окончания работ и равен нулю

$$t_{рн}(0) = t_{по}(0) = 0$$

где $t_{рн}^{(0)}$ – ранний срок начала работ исходного события; $t_{по}^{(0)}$ – поздний срок окончания исходного события.

Ранний срок начала работ $t_{рн}^{(j)}$ совершения j -го события определяется по формуле

$$t_{рн}(j) = t_{рн}(i) + T_{(i-j)}$$

где $T_{(i-j)}$ – продолжительность работы, ч; $t_{рн}^{(i)}$ – ранний срок начала предшествующего события, ч.

Если событие имеет несколько предшествующих путей, а следовательно, несколько предшествующих событий i , то для оценки раннего срока совершения j -го события следует выбирать максимальный из предшествующих путей.

$$t_1^{рн} = t_0^{рн} + t_{01} = 0 + 2 = 2;$$

$$t_2^{рн} = t_0^{рн} + t_{02} = 0 + 3 = 3;$$

$$t_3^{рн} = \max_{i=1,2}(2 + 2, 3 + 3) = 6;$$

$$t_4^{рн} = t_2^{рн} + t_{24} = 3 + 2 = 5;$$

$$t_5^{рн} = \max_{i=3,4}(6 + 3, 5 + 7) = 12;$$

$$t_6^{рн} = \max_{i=3,4,5}(6 + 2, 12 + 6, 5 + 5) = 18.$$

Найдем ранние сроки окончания работ. Этот срок является наиболее ранним (минимальным) из возможных моментов окончания работы при заданной

$$t_{ij}^{po} = t_i^{pn} + t_{ij}$$

$$t_{01}^{po} = t_0^{pn} + t_{01} = 0 + 2 = 2$$

$$t_{02}^{po} = t_0^{pn} + t_{02} = 0 + 3 = 3$$

$$t_{13}^{po} = t_1^{pn} + t_{13} = 2 + 2 = 4$$

$$t_{23}^{po} = t_2^{pn} + t_{23} = 3 + 3 = 6$$

$$t_{24}^{po} = t_2^{pn} + t_{24} = 3 + 2$$

$$t_{35}^{po} = t_3^{pn} + t_{35} = 6 + 3 = 9$$

$$t_{45}^{po} = t_4^{pn} + t_{45} = 5 + 7 = 12$$

$$t_{36}^{po} = t_3^{pn} + t_{36} = 6 + 2 = 8$$

$$t_{46}^{po} = t_4^{pn} + t_{46} = 5 + 5 = 10$$

продолжительности работ, т.е. $t_{56}^{po} = t_5^{pn} + t_{56} = 12 + 6 = 18$.

Поэтому

Рассчитаем обратный проход. Если $i=n$, то $t_n^{po} = t_n^{pn}$ и является отправной точкой обратного прохода.

Т.о. $t_j^{po} = \min_j(t_j^{po} + t_{ij})$ для всех работ (i,j)и поэтому

$$t_6^{po} = t_6^{pn} = 18,$$

$$t_5^{po} = t_6^{pn} + t_{56} = 18 - 6 = 12 ;$$

$$t_4^{po} = \min(18 - 5, 12 - 7) = 5 ;$$

$$t_3^{po} = \min(18 - 2; 12 - 3) = 9;$$

$$t_2^{po} = \min(5 - 2, 9 - 3) = 3;$$

$$t_1^{po} = t_3^{po} - t_{13} = 9 - 2 = 7;$$

$$t_0^{po} = \min(7 - 2, 3 - 3) = 0$$

Вычислим поздние сроки начала работ. Эти сроки являются наиболее поздними (максимальными) из допустимых моментов начала данной работы, при которых еще возможно выполнение всех последующих работ в установленный срок, т.е.

$$t_{ij}^{pn} = t_j^{po} - t_{ij}.$$

Следовательно:

$$t_{01}^{пн} = t_1^{по} - t_{01} = 7 - 2 = 5$$

$$t_{02}^{пн} = t_2^{по} - t_{02} = 3 - 3 = 0$$

$$t_{13}^{пн} = t_3^{по} - t_{13} = 9 - 2 = 7$$

$$t_{23}^{пн} = t_3^{по} - t_{23} = 9 - 3 = 6$$

$$t_{24}^{пн} = t_4^{по} - t_{24} = 5 - 2 = 3$$

$$t_{35}^{пн} = t_5^{по} - t_{35} = 12 - 3 = 9$$

$$t_{45}^{пн} = t_5^{по} - t_{45} = 12 - 7 = 5$$

$$t_{36}^{пн} = t_6^{по} - t_{36} = 18 - 2 = 16$$

$$t_{46}^{пн} = t_6^{по} - t_{46} = 18 - 5 = 13$$

$$t_{56}^{пн} = t_6^{по} - t_{56} = 18 - 6 = 12$$

Существует два вида резерва времени: **полный резерв** (r_n) и **свободный резерв** $r_{св}$. Полный резерв показывает, на сколько может быть увеличена сумма продолжительности всех работ относительно критического пути, представляет разность между максимальным отрезком времени, в течение которого может быть выполнена работа, и его продолжительность рассчитывается по формуле $r_{пij} = t_{ij}^{пн} - t_i^{пн}$.

Результаты вычислений:

$$r_{п01} = t_{01}^{пн} - t_0^{пн} = 5 - 0 = 5$$

$$r_{п02} = t_{02}^{пн} - t_0^{пн} = 0 - 0 = 0$$

$$r_{п13} = t_{13}^{пн} - t_1^{пн} = 7 - 2 = 5$$

$$r_{п23} = t_{23}^{пн} - t_2^{пн} = 6 - 3 = 3$$

$$r_{п24} = t_{24}^{пн} - t_2^{пн} = 3 - 3 = 0$$

$$r_{п35} = t_{35}^{пн} - t_3^{пн} = 9 - 6 = 3$$

$$r_{п45} = t_{45}^{пн} - t_4^{пн} = 5 - 5 = 0$$

$$r_{п36} = t_{36}^{пн} - t_3^{пн} = 16 - 6 = 10$$

$$r_{п46} = t_{46}^{пн} - t_4^{пн} = 13 - 5 = 8$$

$$r_{п56} = t_{56}^{пн} - t_5^{пн} = 12 - 12 = 0$$

Свободный резерв времени – максимальное время, на которое можно отсрочить начало или увеличить продолжительность работы, при условии,

что все события наступают в ранние сроки: $r_{свиj} = t_j^{пн} - t_i^{пн} - t_{ij}$. вычислим свободные резервы для всех работ:

$$r_{св01} = t_1^{пн} - t_0^{пн} - t_{01} = 2 - 0 - 2 = 0$$

$$r_{св02} = t_2^{пн} - t_0^{пн} - t_{02} = 3 - 0 - 3 = 0$$

$$r_{св13} = t_3^{пн} - t_1^{пн} - t_{13} = 6 - 2 - 2 = 2$$

$$r_{св23} = t_3^{пн} - t_2^{пн} - t_{23} = 6 - 3 - 3 = 0$$

$$r_{св24} = t_4^{пн} - t_2^{пн} - t_{24} = 6 - 3 - 2 = 2$$

$$r_{св35} = t_5^{пн} - t_3^{пн} - t_{35} = 13 - 6 - 3 = 4$$

$$r_{св45} = t_5^{пн} - t_4^{пн} - t_{45} = 12 - 5 - 7 = 0$$

$$r_{св46} = t_6^{пн} - t_4^{пн} - t_{46} = 18 - 5 - 5 = 8$$

$$r_{св56} = t_6^{пн} - t_5^{пн} - t_{56} = 18 - 12 - 6 = 0$$

Следует отметить, что критические работы имеют нулевой полный резерв времени, при этом свободный резерв также должен быть равен нулю.

Используя результаты вычислений при прямом и обратном проходах, можно определить работы критического пути. Работа (i,j) принадлежит критическому пути. Если для нее выполняются условия

$$t_i^{пн} = t_i^{по}, t_j^{пн} = t_j^{по}, t_j^{пн} - t_i^{пн} = t_j^{по} - t_i^{по} = t_{ij}$$

В рассматриваемом примере критический путь включает в себя работы (0,2), (2,4), (4,5), (5,6) с общим временем выполнения работ 18 недель.

Все расчеты запишем в таблицу.

Работа (i,j)	Продолжительность t_{ij}	Ранее		Позднее		Полный резерв	Свободный резерв
		Начало $t_j^{пн}$	Окончание $t_{ij}^{по}$	Начало $t_{ij}^{пн}$	Окончание $t_i^{по}$	r_n	$r_{св}$
	2		4		6	7	8
	2		4		6		
0,1)	3		2		7	5	0
(0,2)	2		3		3	0	0
(1,3)	3	0	4	0	9	5	2
(2,3)	2	2	6	7	9	3	0
(2,4)	3	3	5	6	5	0	1
(3,5)	7	3	9	3	12	3	4
(4,5)	2	6	12	8	9	0	0
(3,6)	5	5	10	5	18	10	11
(4,6)	6	6	18	16	18	8	8
(5,6)		12		12		0	0

При небольшом числе работ критические операции могут быть найдены перебором возможных путей.

Задания для самостоятельного выполнения:

Решить задачи по вариантам. На основе данных таблицы:

1. Построить сетевой график.
2. Выписать все частичные и полные пути.
3. Определить критический путь и задачи, лежащие на критическом пути.
4. Определить оптимальный путь.
5. Найти ранние сроки начала и окончания работ.
6. Найти поздние сроки начала и окончания работ.
7. Определить три типа резервов.
8. Все данные вынести в таблицу.
9. Ответить на контрольные вопросы.

Вариант 1

Руководитель проекта разработал следующий перечень работ:

Работа	Непосредственно предшествующая работа	Время выполнения
A	-	4
B	-	6
C	-	5
D	B	2
E	A	9
F	B	4
G	C, D	8
H	B, E	3
I	F, G	5
J	H	7

Вариант 2

Проект пуско-наладки компьютерной системы состоит из восьми работ.

Непосредственно предшествующие работы и продолжительность выполнения работ показаны ниже.

Работа	Непосредственно предшествующая работа	Время выполнения
A	-	3
B	-	6
C	A	2
D	B, C	5
E	D	4
F	E	3

G	B, C	9
H	F, G	3

Вариант 3

Рассмотрите следующую сеть проекта (продолжительность работ показана в неделях):

Работа	Непосредственно предшествующая работа	Время выполнения
A	-	5
B	-	3
C	A	7
D	A	6
E	B	7
F	D, E	3
G	D, E	10
H	C, F	8

Вариант 4

Учебное заведение разрабатывает новую программу повышения квалификации преподавателей. Желательно, чтобы эту программу можно было реализовать в наиболее сжатые сроки. Существуют взаимосвязи между дисциплинами, которые необходимо отразить, составляя расписание занятий по программе. Дисциплины и их взаимосвязь указаны в следующей таблице.

Дисциплина	Непосредственно предшествующая дисциплина	Время изучения в днях
A	-	4
B	-	6
C	A	2
D	A	6
E	C, B	3
F	C, B	3
G	D, E	5

Вариант 5

В таблице показаны этапы покупки нового автомобиля.

Работа	Предшествующая работа	Длительность (дни)
A: Принятие окончательного решения о покупке автомобиля	-	3

В: Поиск потенциального покупателя имеющегося автомобиля	А	14
С: Составление списка желаемых моделей машин	А	1
Д: Исследование желаемых моделей	С	3
Е: Консультации у автомехаников	С	1
F: Сбор рекламных материалов продавцов автомобилей	С	2
G: Обобщение полученной информации	D,E,F	1
Н: Выбор трех наиболее подходящих моделей	G	1
I: Знакомство "в натуре" с выбранными моделями	Н	3
J: Сбор финансовой информации	Н	2
К: Выбор одного автомобиля	I, J	2
L: Выбор продавца автомобиля	К	2
M: Выбор автомобиля желаемого цвета	L	4
N: Повторная дорожная проверка выбранной модели	L	1
О: Покупка нового автомобиля	B,M,N	3

Варианты 6

Работы, которые необходимо выполнить при реализации проекта переустройства рынка, их взаимосвязь и время выполнения каждой из работ указаны в следующей таблице.

Р абота	Содержание работы	Непосредс твенно предшествующая работа	Время выполнения (недель)
А	Подготовить архитектурный проект	-	5
В	Определить будущих арендаторов	-	6
С	Подготовить проспект для арендаторов	А	4
Д	Выбрать подрядчика	А	3

E	Подготовить документы для получения разрешения	A	1
F	Получить разрешение на строительство	E	4
G	Осуществить строительство	D, F	14
H	Заклучить контракты с арендаторами	B, C	12
I	Вселить арендаторов в павильоны	G, H	2

Вариант 7 Рассмотрите

следующую сеть проекта:

Работа	Непосредственно предшествующая работа	Время выполнения
A	-	3
B	-	8
C	A	6
D	A	6
E	B	9
F	D, E	3
G	D, E	7
H	C, F	8

Вариант 8

В таблице приведены работы, выполняемые при строительстве нового каркасного дома.

Работа	Предшествующие работы	Длительность (дни)
A - Очистка строительного участка	—	1
B - Завоз оборудования	—	2
C - Земляные работы	A	1
D - Заливка фундамента	C	2
E - Наружные сантехнические работы	B,	6
F - Возведение каркаса дома	D	10
G - Прокладка электропроводки	F	3

Н - Создание перекрытий	G	1
I - Создание каркаса крыши	F	1
J - Внутренние сантехнические работы	E,	5
K - Покрытие крыши	I	2
L - Наружные изоляционные работы	F,	1
M - Вставка окон и наружных дверей	F	2
N - Обкладка дома кирпичом	L	4
O - Штукатурка стен и потолков	G	2
P - Облицовка стен и потолков	O	2
Q - Изоляция крыши	I	1
R - Окончание внутренних отделочных работ	P	7
S - Окончание наружных отделочных работ	I	7
T - Ландшафтные работы	S	3

Вариант 9

Работы, которые следует выполнить перед началом строительства новой библиотеки, представлены ниже. Продолжительность работ показана в неделях.

Работа	Содержание работы	Непосредственно предшествующая работа	Время выполнения (недель)
A	Определить место строительства	-	6
B	Разработать первоначальный проект	-	8
C	Получить разрешение на строительство	A, B	12
D	Выбрать архитектурную мастерскую	C	4
E	Разработать смету затрат на строительство	C	6

F	Разработать проект строительства	D, E	15
G	Получить финансирование	E	12
H	Нанять подрядчика	F, G	8

Вариант 10

В таблице приведены этапы выполнения работ по замене линии электропередач.

Работа	Предшествующие работы	Длительность (дни)
A: Определение объема работ	-	1
B: Извещение пользователей о временном отключении электросети	A	0,5
C: Подвозка материалов и оборудования	A	1
D: Предварительные работы	A	0,5
E: Заготовка опор и материалов	C, D	3
F: Развозка опор	E	3,5
G: Определение нового местоположения опор	D	0,5
H: Разметка местоположения опор	G	0,5
I: Земляные работы для установки новых опор	H	3
J: Установка новых опор	F, I	4
K: Ограждение старой линии	F, I	1
L: Прокладка новых проводов	J, K	2
M: Обустройство новой линии	L	2
N: Натяжка проводов	L	2
O: Подрезка деревьев	D	2
P: Отключение старой электролинии	B, M, N, O	0,1
Q: Подключение новой электролинии	P	0,5
R: Уборка территории	Q	1

S: Удаление проводов старой линии	Q	1
T: Удаление опор старой линии	S	2
U: Возврат материалов и оборудования	R, T	2

Вариант 11

Компания готовит бюджет производства нового изделия. В таблице представлены этапы подготовки бюджета и их длительность.

Работа	Предшествующие работы	Длительность (дни)
A: Прогнозирование объема продаж	—	10
B: Изучение рынка конкурирующих товаров	—	7
C: Доводка изделия	A	5
D: Подготовка производственного плана	C	3
E: Оценка стоимости производства	D	2
F: Определение отпускной цены	B, E	1
G: Подготовка бюджета	E, F	14

Вариант 12

В таблице приведена последовательность работ по разработке и производству станков.

Работа	Предшествующие работы	Длительность
A – составление сметы затрат	–	3
B – согласование оценок	A	6
C – покупка собственного оборудования	B	1
D – подготовка конструкторских проектов	B	2
E – строительство основного цеха	D	10
F – монтаж оборудования	C, E	5
G – испытание оборудования	F	4

Н – определение типа модели	D	9
I – проектирование внешнего корпуса	D	7
J – создание внешнего корпуса	H,I	6
K – конечная сборка	G,J	3
L – контрольная проверка	K	7

Вариант 13

Проект «Внедрение бухгалтерской системы» для небольшой бухгалтерии, содержащей порядка 10 рабочих мест.

Работа	Предшествующие работы	Длительность
A – Начало проекта	–	-
B – Выбор системы	A	60
C – Приобретение программного обеспечения	B	25
D – Составление проекта сети	B	15
E – Приобретение компьютеров и сетевого оборудования	B	30
F – Обучение администратора и программиста	D	20
G – Монтаж локальной сети	D,E	15
H – Установка ПО на компьютеры	C,E	70
I – Установка сетевого ПО, настройка сети	F,G,H	30
J – Ввод начальных данных в информационную базу	I	90
K – Обучение персонала	I	20
L – Передача в эксплуатацию	J,I	5
M - Конец проекта	-	-

Вариант 14

В таблице приведены данные для построения сетевого графика для планирования производства некоторого изделия.

Работа	Предшествующие работы	Длительность
--------	-----------------------	--------------

А – Разработка технического задания	–	5
В – Конструирование оснастки	-	8
С – Разработка электро-схемы	А	3
Д – Разработка сборочных чертежей	А	8
Е – Разработка технологии сборки	С	2
F – Изготовление оснастки	В	4
G – Монтаж электро-схемы	С	7
Н – Сборка изделия	D,E	5
I – Окраска изделия	F,G	2

Вариант 15

Перечень выполняемых проектных работ по сбыту продукции.

Работа	Предшествующие работы	Длительность (дни)
А: Обоснование цели проекта	-	2
В: Проведение маркетинговых исследований	А	5
С: Разработка технических условий	А	3
Д: Эскизное проектирование	А	4
Е: Выбор поставщиков ресурсов	В	2
F: Фиктивная работа	С	0
G: Техническое проектирование	Д	5
Н: Расчет потребности ресурсов	Е	2
I: Рабочее проектирование	G	10
J: Закупка производственных ресурсов	Н	10
К: Изготовление деталей	I	8
L: Сертификация деталей	Н	2

М: Согласование сроков поставки	I	3
N: Разработка технологии сборки	J,K	3
O: Сборка изделия	N	11
P: Отправка продукции потребителям	O	5

Вариант 16

Перечень работ по привлечению клиентов банка

Работа	Предшествующие работы	Длительность (дни)
A: Разработка плана мероприятий	-	2
B: Поиск идеи депозитного вклада	A	30
C: Разработка системы критериев по выбору идеи	A	21
D: Выбор идеи	B, C	2
E: Техничко-экономическое обоснование идеи	D	2
F: Составление проекта договора банка с вкладчиком по депозитному счету	E	5
G: Представление проекта договора руководству банка, его рассмотрение и утверждение	F	7
H: Заказ в типографию на изготовление бланка договора и его выполнение	G	20
I: Заказ на рекламу в печати и его выполнение	G	14
J: Заказ на изготовление стенда и его изготовление	G	7
K: Получение типографского договора	H	1
L: Оформление зала к приему клиентов	I, J	2

М: Семинар с работниками банка	K, L	1
N: Начало работы с клиентами	M	0

Контрольные вопросы:

1. Что представляет собой сетевое планирование?
2. Дайте определения понятиям событие, работа?
3. Что представляет собой действительная работа? Что такое фиктивная работа?
4. Назовите правила построения классических сетевых графиков?
5. Что такое путь? Что представляет собой полный путь? Критический путь?
6. Назовите временные характеристики сетевого графика?
7. Что такое резерв времени работы? Перечислите разновидности резервов времени работ?

Практическая работа №20.

Создание диаграмм UML в программе MS VISIO.

Цель: Получение навыков создания UML диаграмм в пакете MSVisio. Научиться осуществлять надстройку программы для редактирования UML диаграмм.

В результате освоения учебной дисциплины обучающийся должен **уметь**:

- ☐ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;
- ☐ использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения.

В результате освоения учебной дисциплины обучающийся должен **знать**:

- ☐ типы организационных структур;
- ☐ требования к проектируемой системе;
- ☐ модели жизненного цикла информационной системы, методы проектирования информационной системы;
- ☐ технологии проектирования информационной системы, оценку и управление качеством информационной системы.

Теоретическая часть

UML (Unified Modeling Language) - унифицированный язык моделирования – это язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели

системы, называемой *UML моделью*. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем.

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Рассмотрим более подробно некоторые виды диаграмм.

Диаграммы прецедентов.

Диаграмма прецедентов (Диаграмма вариантов использования, Use case diagram) — диаграмма, на которой отражены отношения, существующие между участниками и вариантами использования.

Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Каждый прецедент(use case) характеризует определенный тип использования системы участником; у каждого прецедента есть имя, и он может иметь текстовое пояснение.

Участник (актёр, actor) - представляет кого угодно (что угодно) не относящегося к системе, это тот, кто использует систему и реагирует на её действия (человек, железо, время, другая система); единственное действие участника это приём и передача информации системе.

Ограничения (boundary, граница системы) - это классификатор (система/подсистема/класс), функциональность которого мы описываем с помощью прецедентов; ограничения определяют границы системы/подсистемы.

Отношения (relationships) - ассоциация между участником и прецедентом показывает направление информации между внешним элементом и прецедентом.

Если зависимость между прецедентами содержит стереотип <<включает>> (include), значит, первый прецедент включает действия второго (в Visio используется стереотип <<uses>>). Так же присутствуют связи <<расширяет>> - при выполнении расширяемого прецедента (*стрелка указывает на него*) выполнения расширителя не обязательно, но возможно, <<обобщение>> (аналогично наследованию классов). Так же прецеденты могут быть соединены ассоциациями (линия или стрелка), имя которых должно раскрывать суть взаимодействия прецедентов. **Пример диаграммы прецедентов**

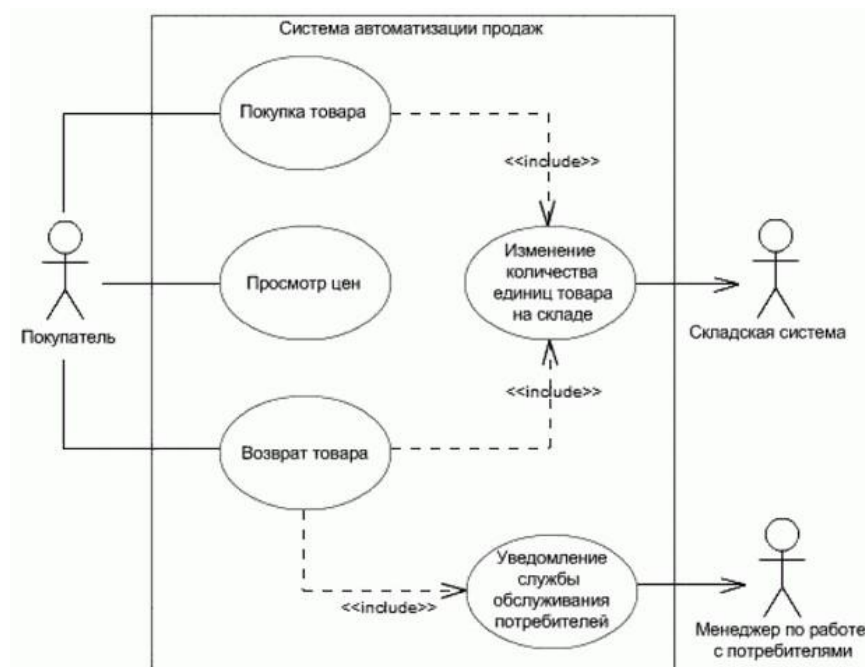


Рис. 1 Пример диаграммы прецедентов

Диаграммы последовательности.

Диаграммы последовательности (sequence diagram) отображают динамику взаимодействия объектов во времени. Объекты на диаграмме располагаются слева направо. Время идет сверху вниз.

Основные элементы диаграммы последовательности.

Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности от самой верхней ее части до самой нижней.

Фокус управления (focus of control).. Чтобы явно выделить подобную активность объектов, в языке UML применяется специальное понятие, получившее название фокуса управления. Фокус управления изображается в форме вытянутого узкого прямоугольника, верхняя сторона которого обозначает начало получения фокуса управления объектом (начало активности), а его нижняя сторона - окончание фокуса управления (окончание активности). Прямоугольник располагается ниже обозначения соответствующего объекта и может заменять его линию жизни, если на всем ее протяжении он является активным.

Сообщения. В UML каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой. Сообщение представляет собой законченный фрагмент информации, который отправляется одним объектом другому. Прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено. В языке UML различаются несколько разновидностей сообщений, каждое из которых имеет свое графическое изображение:

- первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки всегда соприкасается с фокусом

управления или линией жизни того объекта-клиента, который инициирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. Принимающий объект, как правило, получает фокус управления, становясь активным;

- вторая разновидность сообщения используется для обозначения простого потока управления. Каждая такая стрелка указывает на выполнение одного шага потока. Такие сообщения, обычно, являются асинхронными, то есть могут возникать в произвольные моменты времени. Передача такого сообщения, как правило, сопровождается получением фокуса управления, принявшим его объектом;

- третья разновидность явно обозначает асинхронное сообщение между двумя объектами в некоторой процедурной последовательности. Примером такого сообщения может служить прерывание операции при возникновении исключительной ситуации. В этом случае информация о такой ситуации передается вызывающему объекту для продолжения процесса дальнейшего взаимодействия;

- четвертая разновидность сообщения используется для возврата из вызова процедуры. Примером может служить простое сообщение о завершении некоторых вычислений без предоставления результата расчетов объекту-клиенту.

Практическая работа №21.

Диаграммы классов.

Цель: Получение навыков создания UML диаграмм в пакете MSVisio. Научиться осуществлять надстройку программы для редактирования UML диаграмм.

Диаграммы классов.

Диаграмма классов описывает статическую структуру системы. На ней отображены классы, их методы и атрибуты, а так же связи между ними.

Основной элемент диаграммы классов – это класс. Он обозначается прямоугольником, горизонтально разделенным на 3 части. В верхней записывается имя класса, в центральной атрибуты, в нижней – методы. Если требуется изобразить интерфейсный класс, то он будет содержать только 2 поля – для имени и методов, так как у такого класса нет атрибутов.

Классы могут находиться в следующих типах отношений:

- Ассоциация показывает, как объекты одного класса связаны с объектами другого. Обозначаются линиями, идущими от одного класса к другому. Самыми распространенными являются однонаправленные и двунаправленные ассоциации. Иногда концах линий пишут мультипликаторы, чтобы показать количество объектов, участвующих в ассоциации (1 .. 1, 1 .. n, и т.д.).

- Агрегация — это разновидность ассоциации при отношении между целым и его частями. Как тип ассоциации агрегация может быть именованной. Агрегация встречается, когда один класс является коллекцией или контейнером других. Если контейнер будет уничтожен, то его содержимое — нет.

- Композиция — более строгий вариант агрегации. Известна также как агрегация по значению. Композиция имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов. Если контейнер будет уничтожен, то всё его содержимое будет также уничтожено.

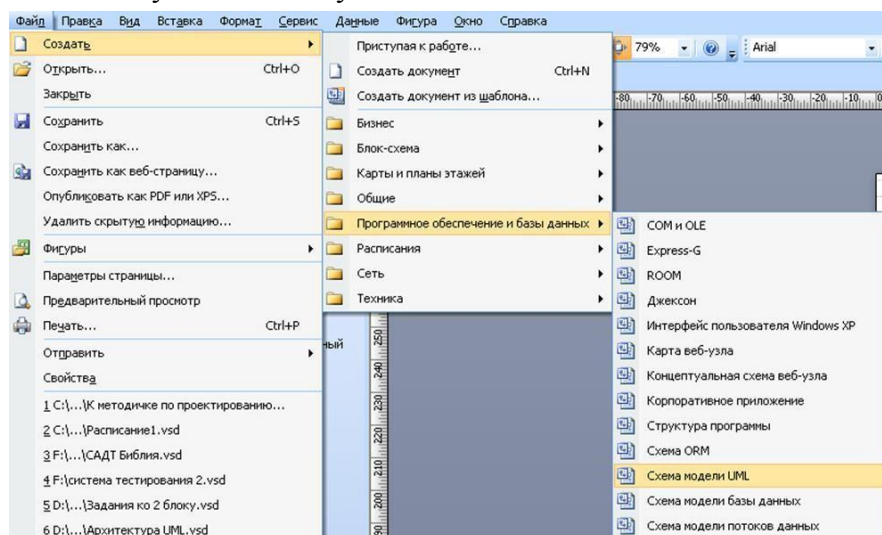
- Обобщение на диаграммах классов

используется, чтобы показать связь между классом-родителем и классом-потомком. Оно вводится на диаграмму, когда возникает разновидность какого-либо класса (например: животное - рептилия), а также в тех случаях, когда в системе обнаруживаются несколько классов, обладающих сходным поведением

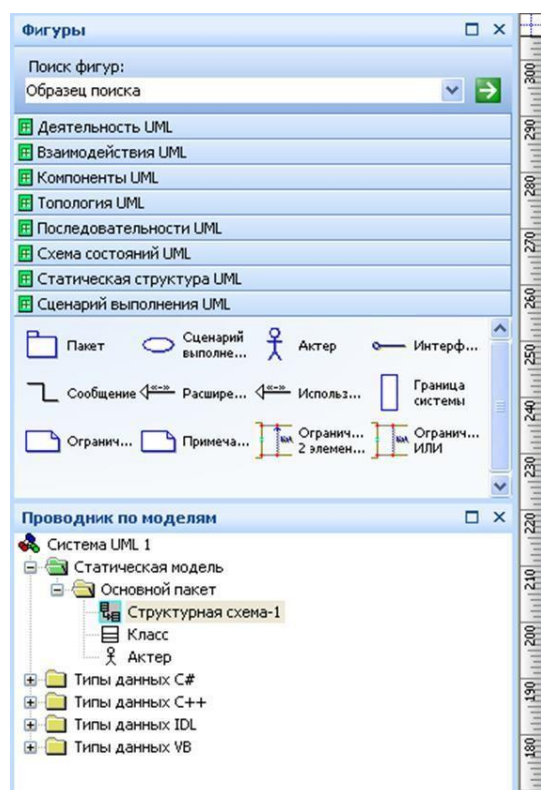
Практическая часть

Visio - решение для построения диаграмм от Microsoft. Visio помогает преобразовать технические и бизнес-концепции в визуальную форму. И действительно, этот пакет из семейства Microsoft Office предназначен исключительно для рисования диаграмм. Visio имеет некоторые дополнительные возможности, но все же, *по большей мере* - это только средство для иллюстрирования документов MS Office.

Создание диаграмм UML в программе MSVISIO Для того, чтобы включить надстройку для редактирования UML, нужно создать соответствующий документ Visio:



После запуска надстройки появиться подменю UML, а также автоматически откроются библиотеки для различных диаграмм UML и проводник по модели UML:



Поддержка UML в Visio заключается в поддержке всех основных элементов UML, возможность редактирования их свойств, механизмов добавления, стереотипов, а также проводники по моделям.

Проводник по моделям показывает в иерархическом виде все элементы, добавленные на любые диаграммы. Поскольку диаграммы UML представляют собой связное описание одной информационной системы, то на различных диаграммах могут использоваться одни и те же объекты. Если на какой-либо диаграмме нужно разместить элемент, который уже создан, то этот элемент нужно перетаскивать из проводника по моделям.

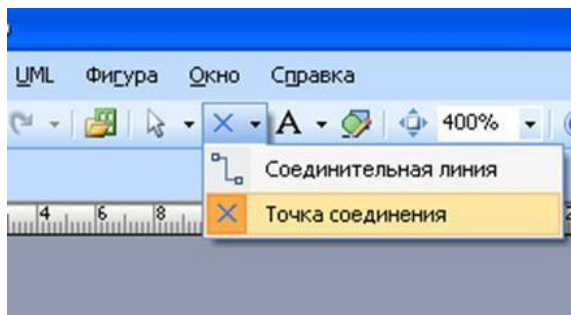
Для правильной работы проводником по моделям нужно иметь в виду следующие моменты. Если добавить элемент UML на соответствующую ему диаграмму, то этот элемент автоматически появится в модели. Если же удалить элемент с диаграммы, то в модели этот элемент останется и удалить его можно только вручную. Если добавить элемент на неподходящую, то этот элемент в модели не появится. В связи с этим, во избежание путаницы, имеет смысл все операции создания, удаления и редактирования элементов UML проводить в проводнике по моделям.

Важной частью работы с элементами UML является редактирование их свойств. Вызвать окно свойств можно либо двойным кликом по элементу, либо через контекстное меню элемента в проводнике по моделям.

В MSVisio существует возможность указывать какие свойства элемента нужно отображать на диаграмме. Для этого нужно на элементы UML вызвать контекстное меню «параметры отображения фигуры» и поставить галочки против тех параметров, которые должны отображаться.

При рисовании диаграмм необходимо правильно отобразить точки соединения, к которым присоединяются соединительные линии.

Для того, чтобы добавить точку соединения необходимо ее выбрать в меню:



Для того, чтобы поместить точку соединения нужно нажать клавишу “CTRL” или кликнуть по тому месту, где должна находиться точка.

Практическая работа №22. Программа MS VISIO.

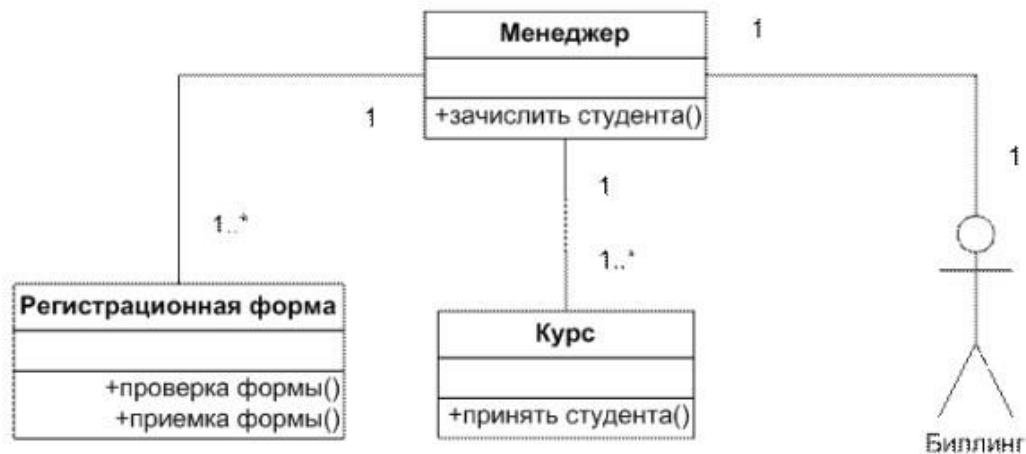
Цель: Получение навыков создания UML диаграмм в пакете MS Visio. Научиться осуществлять надстройку программы для редактирования UML диаграмм.

Задание к работе:

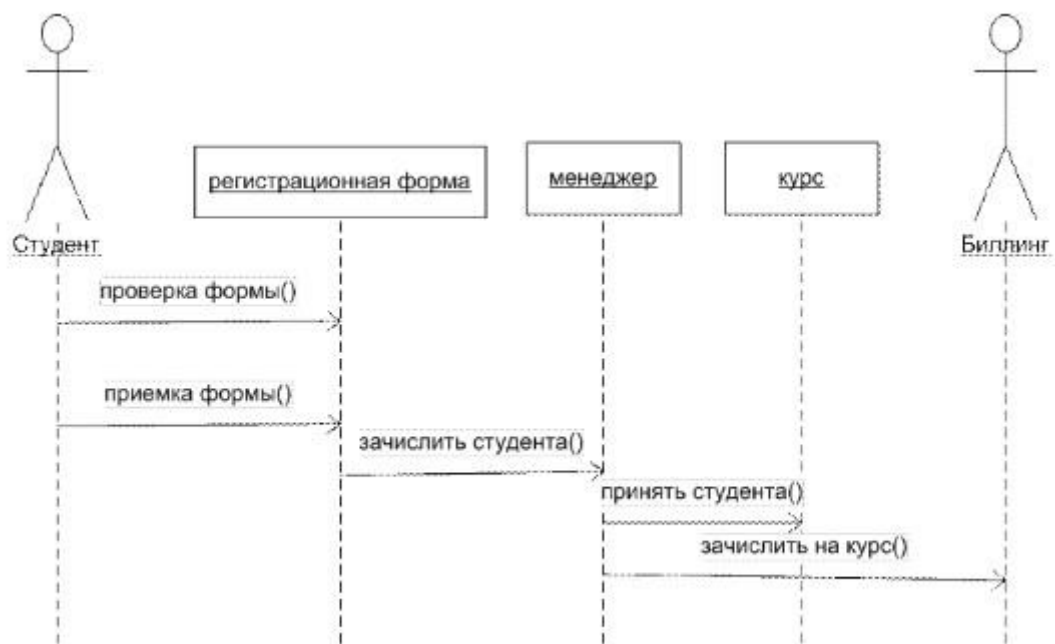
1. Нарисовать диаграмму прецедентов по следующему образцу:



2. Нарисовать диаграмму классов по следующему образцу:



3. Создайте диаграмму последовательностей по следующему образцу:



Для изображения объектов используется элемент «линия жизни» из библиотеки «Последовательности UML». На этой диаграмме между объектами устанавливаются не отношения, а посылаются сообщения, которые должны задаваться с помощью соответствующего элемента.

Обратите внимание на привязку элементов (должно быть красных, не присоединенных концов). Надо установить привязку объектов к их классам (классификатор). Если для объекта установить классификатор актера, то изображение автоматически изменится на изображение актера. Сообщения нужно привязать к операциям, которые есть объекта. По проводнику моделям проверьте, что добавлены все нужные операции к классам, и не добавлено лишних операций. Проверьте, что операции отобразились на диаграмме классов.

4. Создайте стереотип «extend» отношения зависимость, добавьте на диаграмму прецедентов прецедент «создание курса», который расширяет прецедент «управление курсом».

Контрольные вопросы:

1. Что такое UML?
2. Что представляет собой диаграмма прецедентов?

3. Что представляет собой диаграмм последовательности? Основные элементы диаграммы последовательности.

4. Что представляют собой диаграммы классов? В каких типах отношений могут находиться классы?

5. Для чего нужен проводник по моделям?

6. Как разместить один и тот же элемент на различных диаграммах?

7. Как связать на рисунке сущность и отношение UML?

8. Какое отношение по умолчанию используется в Visio?

9. Назовите несколько свойств класса.

Практическая работа № 23

Использование XML-технологий.

Цель: Изучение возможностей и назначение языка XML. Получение навыков использования XML-технологии для организации хранения информации. Изучение способов перевода текстовых и табличных данных в формат XML.

В результате освоения учебной дисциплины обучающийся должен **уметь:**

☐ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;

☐ использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения.

В результате освоения учебной дисциплины обучающийся должен **знать:**

☐ требования к проектируемой системе;

☐ модели жизненного цикла информационной системы, методы проектирования информационной системы; технологии проектирования информационной системы, оценку и управление качеством информационной системы.

Теоретическая часть

Язык разметки XML – одна из наиболее динамично завоевывающих внимание разработчиков и пользователей Интернеттехнологий. Международная организация W3C утвердила спецификацию "Extensible Markup Language(XML) 1.0" еще в начале февраля 1998 года, но только теперь становится ясно, для чего этот самый XML употребить, и каким именно образом.

Дело в том, что сам по себе язык XML ничего особенного не представляет, и никаких задач решить не может. Можно создать структурированный документ или хранить небольшие количества информации, но, во-первых, ничего нового в этом нет, а, во-вторых, результат ваших трудов будет понятен только вам, никакой браузер этот документ понять не сможет. Это может легко привести к ошибкам в интерпретации и отображении информации. На сегодняшний день вокруг XML уже создано огромное количество технологий и, если можно так выразиться, различной "оснастки". Все они имеют одну простую и ясную цель – сделать XML применимым в реальных задачах. Поэтому говорить имеет смысл не об XML, а о способах и технологиях его преобразования и использования, таких, как XSL/XSLT, XPath, XDR, XML DOM и так далее.

Области примененияXML-документов

XML-документы обычно длиннее замещаемых ими бинарных форматов. Они требуют большей полосы пропускания сети, больше места для хранения, и уж точно больших затрат процессорного времени. Разбор XML медленнее, чем разбор оптимизированных бинарных форматов, да к тому же требует больше памяти. Но у XML есть несколько важных достоинств:

1. XML позволяет разработчику создать собственную разметку структуры хранимой информации.

2. Разбор XML хорошо стандартизирован и реализован большим количеством производителей ПО, что позволяет извлечь информацию из XML-документов практически повсеместно.

3. В стандарт XML включена поддержка кодовых страниц Unicode, что упрощает создание многоязычных документов.

4. Приложения могут использовать XML-парсеры для проверки структуры документов, а при использовании схем – и типов данных. Это может значительно упростить разбор строго структурированных документов, снимая с программиста задачу проверки правильности документа.

5. XML – это текстовый формат, то есть читаемый, легко документируемый и, иногда, более простой в отладке. Хотя на сегодня отладочные средства для ряда XML-технологий пребывают в зачаточном состоянии.

6. Для работы с XML создано множество средств на самых разных платформах, что делает использование XML более простым по сравнению с бинарными форматами при обмене сложными информационными потоками.

7. XML-документы могут использовать значительную часть инфраструктуры, созданной для HTML, включая протокол HTTP и браузеры.

Можно выделить множество задач, связанных с созданием и обработкой структурированной информации, для решения которых может использоваться XML:

- На базе XML создана технология SOAP, позволяющая осуществлять программные вызовы методов объектов через HTTP в текстовом формате. Говоря коротко, это один из вариантов маршалинга, но не в компактную бинарную форму, а в текстовый формат с большим количеством метаданных. Это может показаться бредом, но в условиях набитого брандмауэрами Интернета это единственный способ вызывать методы объектов.

- При разработке сложных информационных систем, с большим количеством приложений, связанных потоками информации и самой различной структурой. В этом случае XML-документы выполняют роль универсального формата для обмена информацией между отдельными компонентами большой программы.

- Язык XML позволяет описывать данные произвольного типа, в том числе и специализированную информацию, например химические, математические, физические формулы, медицинские рецепты, нотные записи и т.д. Это означает, что XML может служить мощным дополнением к HTML для представления в Web "нестандартной" информации.

- XML-документы могут использоваться в качестве промежуточного формата данных в трехзвенных системах. Обычно схема взаимодействия между серверами приложений и баз данных зависит от конкретной СУБД и диалекта SQL, используемого для доступа к данным.

- Из спецификаций XLink и Xpointer родилась вполне пристойная спецификация XPath, позволяющая создавать краткие и эффективные запросы к структуре документа, ссылаясь как на отдельные элементы документа, так и на их группы с учетом вложенности и значений атрибутов. □ Использование стилевых таблиц (XSL/XSLT) позволяет упростить изменение структуры и преобразование XML- документов, или их преобразование в формат HTML.

- XML может использоваться в обычных приложениях для хранения и обработки структурированных данных в едином формате.

Моделирование данных и XML

Успех любого приложения XML зависит от того, насколько хорошо спроектированы фактически используемые документы XML: они должны быть способны не только нести информацию, которую люди передают друг другу сегодня, но и обладать достаточной гибкостью, чтобы учитывать будущие требования. Для этого необходимо рассмотреть следующие аспекты процесса проектирования:

Моделирование информации (понимание структуры назначения информации, содержащейся в документах);

Проектирование документа (трансляция информационной модели в набор правил или схем для создания фактических документов);

Нотации схем (методы записи проекта документа, чтобы он был доступен как для обрабатывающего его программного обеспечения, так и для пользователя-человека).

Информационная модель — это описание используемой организацией информации, не зависящее от какой бы то ни было информационной технологии и определение таких моментов, как:

- каким образом она структурирована;
- что она означает;
- кому она принадлежит, и кто отвечает за ее своевременность и качество;
- откуда она берется и что происходит с ней в конце.

Моделирование информации имеет большое значение, потому что без модели нет информации, есть только данные. Информационная модель описывает назначение данных.

Любое информационное моделирование преследует две цели, которые не всегда бывает легко сочетать:

- ☐ Получение абсолютно точных определений
- ☐ Эффективная коммуникация с пользователями

Существуют два основных типа информационной модели: статическая и динамическая.

Статическая информационная модель

В *статической* модели описываются допустимые состояния системы, типы объектов в системе, их свойства и связи (вместе с этим определяются их имена). Достичь соглашения по именованию всех объектов очень важно, именно поэтому информационные модели XML иногда называют словарями.

При построении статической информационной модели необходимо пройти следующие этапы:

Этап 1. Идентификация понятий, присвоение им имен и их определение

Этап 2. Организация понятий в иерархию классов

Этап 3. Определение связей, множественности и ограничений

Этап 4. Добавление свойств для конкретизации деталей значений, связанных с объектами

Именование понятий

Для начала нужно составить список понятий, относящихся к системе. Иногда предлагают описать систему на бумаге и выделить все существительные. В любом случае этот этап, как правило, не представляет затруднений.

Далее, и это иногда требует больше времени, надо описать типы объектов. Описание термина должно быть точным, чтобы не возникало разногласий по существу определения. Ценность моделирования в том и заключается, что оно предотвращает появление потенциальных источников непонимания.

По завершении работы, вероятно, получится длинный список типов объектов, и у некоторых будут длинные имена. Следует выбирать такие имена, которые занятые в этом бизнесе люди смогут корректно понять и интерпретировать: дело в том, что они не всегда будут сверяться с написанными определениями.

Помимо именования типов объектов стоит определить также, каким образом можно идентифицировать индивидуальные экземпляры. Возможно, в этом виде бизнеса существует код, который следует знать, или написать его. Надо принимать во внимание, что в схемах кодирования в бизнесе часто обнаруживаются проблемы.

В конце этого этапа мы получим список типов тех объектов с именами и определениями, для которых достигнуто соглашение.

Таксономия

Таксономия — это термин, используемый в информационном моделировании который называют иерархией типов (иногда ее называют еще онтологией). Перечислив и назвав типы объектов, их надо организовать в иерархическую схему классификации. Часто эти иерархические отношения возникают уже на этапе определения типов объектов.

Ключевой здесь является фраза, определяющая принадлежность (в английском языке — *is* или *is kind of*). Написав предложение вида "А есть разновидность В" или "Каждое А есть В", вы определили отношения подтипов в вашей таксономии.

Иногда эти действия называются тестом "is a" ("есть", "представляет собой"). Однако будьте внимательны, поскольку эта конструкция используется также и для описания отношений между конкретным экземпляром и его типом, безопаснее писать этот тест в форме "is a kind of" ("представляет собой разновидность").

Итак, этап 2 сводится к организации типов объектов в иерархию типов.

Поиск связей

После того как объекты названы, в статическом информационном моделировании надо определить *связи*, существующие между ними. Связи (на языке UML они называются ассоциациями) можно показать просто сформулировав их в виде обычных предложений или их можно показать графически в виде диаграммы. Для диаграмм, описывающих связи между объектами, существует большое количество нотаций, каждый может выбрать предпочтительную для себя. Диаграммы следует делать предельно простыми и интуитивно понятными, сосредоточившись на ключевых сообщениях и оставив подробности текстовым документам, которые проще обслуживать.

Существует некая информация, которую надо знать о каждой связи:

- Множественность связи показывает, сколько объектов каждого типа принимает в ней участие.
- Связи типа "один-ко-многим": одна глава содержит много параграфов, один человек покупает много туристических поездок.
- Связи типа "многие-ко-многим" также часто встречаются: один автор может написать несколько книг, но у книги также может быть несколько авторов.

- Связи типа "один-к-одному".

При моделировании информации для окончательного представления XML особенно важным типом связей являются связи включения. Множественность этих связей всегда бывает "один-ко-многим" и "один-к-одному". Хотя четкого правила по поводу того, какие объекты образуют связи включения, не существует, можно иногда использовать правила обычного языка: глава содержит параграфы, курорт содержит отели, а отель содержит посетителей. В языке UML определено две формы связей включения. Первая - это *агрегация*, относительно свободное объединение объектов, позволяющее рассматривать их группу в течение некоторого времени как целое (например, туристическая группа, одни и те же люди могут в разное время входить в разные группы). Вторая форма - *композиция*. Это более строгая форма. Отдельные части целого не могут существовать независимо от него (например, комнаты в отеле не могут существовать независимо от отеля).

Найти подходящие имена для связей бывает нелегко. При записи связей полезно использовать полные фразы типа "отель расположен на курорте" или "человек - это автор книги". Нам не надо использовать эти имена в тегах разметки XML, они присутствуют только в документации на систему.

Итак, мы определили связи, существующие между типами объектов в нашей модели.

Практическая работа № 24.

Взаимодействие с прикладными программами MSOffice, описание свойств.

Цель: Изучение возможностей и назначение языка XML. Получение навыков использования XML-технологии для организации хранения информации. Изучение способов перевода текстовых и табличных данных в формат XML.

Описание свойств

Типы объектов и связи формируют скелет статической информационной модели, свойства можно сравнить с плотью на костях. Свойства представляют собой простые значения, ассоциированные с объектами. У человека можно определить рост, вес, национальность и род занятий; отель имеет определенное количество комнат, этажность и ценовую категорию.

Не следует снова включать связи в список свойств объекта: расположение отеля не является его свойством, если мы уже промоделировали его как связь с курортом.

Главное, что нам надо знать о свойствах, - это тип их данных. Определен ли для них фиксированный диапазон значений, являются ли они числовыми, в каких единицах выражаются? Является ли свойство обязательным и есть ли у него значение по умолчанию?

В конце этапа 4 мы завершили формирование статической информационной модели: получили полное описание типов объектов в системе, их связей друг с другом и их свойств.

Динамическая информационная модель

Динамические модели описывают, что происходит с информацией: примерами таких моделей являются диаграммы рабочих процессов, потоков данных и жизненных циклов объектов. Динамические модели состоят примерно из таких утверждений: "Отделение патологии отправит результаты теста консультанту, отвечающему за

пациента". Динамические модели описывают процесс обмена информацией: данные отправляются из одного места в другое с конкретной целью.

Для построения динамической модели можно воспользоваться различными программными системами (например, диаграммы рабочих процессов и диаграммы потоков данных можно построить, используя программу Ramuseducational).

Модели рабочих процессов

Модели рабочих процессов заостряют внимание на роли людей и организаций в выполнении работы, хранение и обработка информации играют в них вторичную роль. Модель процесса описывает, например, что будет с путешественником, если на отдыхе с ним произойдет несчастный случай. Она определяет, за какие действия отвечает локальный представитель на курорте, агент в стране, где находится курорт и главный офис. В результате становится ясно, кто должен отвечать за организацию медицинской помощи, за перевозку туриста домой и за информирование родственников. Она может описывать различные формы, заполняемые и пересылаемые между участниками, и вообще не привлекать компьютерные системы. Модель процесса обычно фокусирует внимание на ролях, обязанностях и задачах каждого действующего лица системы (actor), а workflow-модель имеет дело с документами, передаваемыми между действующими лицами. Модели потоков данных.

Модели потоков данных

Модели потоков данных очень напоминают предыдущий тип, но здесь основное внимание уделяется информационным, а не бизнес-системам. Эта модель описывает хранилища данных (datastores), где информация находится постоянно (это может быть база данных в компьютере или просто кабинет с папками), процессоры, манипулирующие с этими данными, и потоки данных, передающие данные от одного процессора другому. Она активно использует статическую информационную модель: последняя описывает, что означают такие концепции, как путешественник или отель, но ничего не сообщает о том, где содержится информация. Напротив, из модели потоков данных становится ясно, что информация о туристической поездке находится в базе данных покупок до завершения поездки и оплаты всех счетов, после чего резюме этой информации передается в маркетинговую информационную систему, а все остальное - в архив.

Объектные модели

Объектные модели содержат как динамический, так и статический компоненты. Динамическая или поведенческая часть определения объекта сосредоточена на том, что может делать или делал каждый объект, представляя для этого набор операций или методов, описывающих его действия.

Жизненные циклы объекта

Жизненные циклы объекта (на языке UML это называется линиями жизни объекта) также заостряют внимание на индивидуальных объектах, но придерживаются более целостного подхода. Они описывают, что происходит с объектом на протяжении его жизни: как он создается, какие события с ним происходят, как он реагирует на эти события и какие условия приводят в конце к его разрушению.

Жизненные циклы объекта очень полезны для тестирования завершенности модели. Часто наблюдается тенденция к акцентированию внимания только на некоторых событиях за счет остальных. Пока мы не определим, каким образом каждый объект попадает в систему и как он удаляется из нее, полного понимания не будет.

Варианты использования

Варианты использования (usecases) анализируют выполнение специфических задач пользователя (например, человек, купивший туристическую путёвку, отменяет свой

заказ). Вариант использования напоминает модель процесса, но в общем случае заостряет внимание на деятельности одного конкретного пользователя.

Варианты использования могут быть полезны как на этапе моделирования деловой активности, так и при описании внутреннего поведения информационных систем. Одна из опасностей заключается в смешении двух уровней. Лучше этого не делать, поскольку они представляют интерес для различных аудиторий.

Представленный в виде варианта использования диалог пользовательского интерфейса описывает, какой информацией пользователь обменивается с системой, а не то, как она представлена на экране. Это естественным образом приводит к реализации XML, в которой информационное содержание отделено от особенностей представления.

Диаграммы взаимодействия объектов

Диаграммы взаимодействия объектов позволяют проанализировать обмен сообщениями между объектами на более тонком уровне детализации, чем модель потока данных.

Диаграммы взаимодействия объектов неоценимы, если требуется описать взаимодействие между различными системами. Они позволяют определить, какая информация содержится в каком сообщении. Поскольку эти сообщения написаны на языке XML, диаграммы взаимодействия объектов дают нам контекст, требуемый для начала проектирования структуры XML каждого индивидуального сообщения.

Пример статической информационной модели

Информационную модель данных, которая затем будет трансформирована в XML-документ, можно представить в виде *древовидной* структуры, которая начинается с *корня* и заканчивается *листьями*.

XML-документ должен содержать *корневой элемент*, который является *родительским* для всех остальных элементов. Любой элемент (кроме находящихся на самом нижнем уровне дерева) может иметь вложенные элементы (*дочерние элементы*).

С помощью терминов *родитель*, *дочерний* и *потомок* описываются отношения между элементами в дереве XML документа. *Родители* содержат *дочерние элементы*, а дочерние элементы одного уровня называются *потомками* (*братьями* или *сестрами*).

На рисунке представлен пример статической информационной модели, описывающей отношения между элементами в предметной области «Книжный магазин».

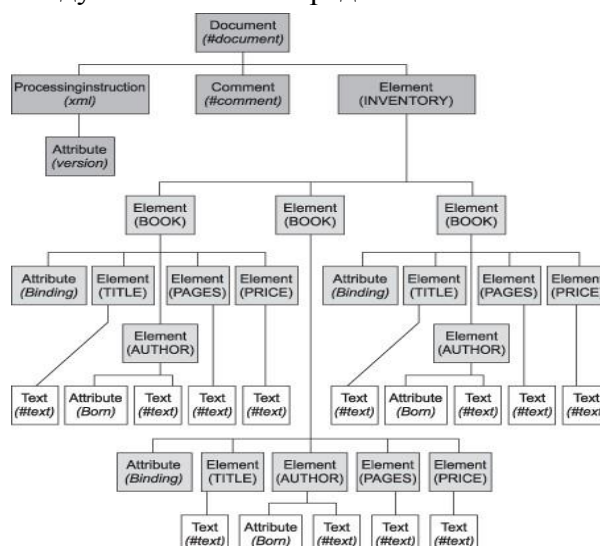


Рис. 1 Пример статической информационной модели

```
<?xml version="1.0" encoding="windows-1251" ?>

<!-- Имяфайла: Inventory Dom.xml -->
<INVENTORY>
<BOOK Binding="mass market paperback">
<TITLE>The Adventures of Huckleberry Finn</TITLE>
<AUTHOR Born="1835">Mark Twain</AUTHOR>
<PAGES>298</PAGES>
<PRICE>$5.49</PRICE>
</BOOK>
<BOOK Binding="trade paperback">
<TITLE>The Marble Faun</TITLE>
<AUTHOR Born="1804">Nathaniel Hawthorne</AUTHOR>
<PAGES>473</PAGES>
<PRICE>$10.95</PRICE>
</BOOK>
<BOOK Binding="hardcover">
<TITLE>Moby-Dick</TITLE>
<AUTHOR Born="1819">Herman Melville</AUTHOR> <PAGES>724</PAGES>
<PRICE>$9.95</PRICE>
</BOOK>
</INVENTORY>
```

Синтаксис XML

Для ограничения тегов в разметке XML, так же как и в HTML, используются угловые скобки: тег начинается со знака "меньше" (<) и завершается знаком "больше" (>). Но необходимо помнить, что в отличие от HTML вся разметка XML чувствительна к регистру символов, это касается как имен тегов, так и значений атрибутов.

Имена

В языке XML все имена должны начинаться с буквы, символа нижнего подчеркивания (_) или двоеточия (:) и продолжаться только допустимыми для имен символами, а именно: они могут содержать только буквы, входящие в секцию букв кодировки Unicode, арабские цифры, дефисы, знаки подчеркивания, точки и двоеточия. Однако имена не могут начинаться со строки *xml* в любом регистре. Имена, начинающиеся с этих символов, зарезервированы для использования консорциумом W3C.

Структура XML- документа

Любой XML-документ состоит из следующий частей:

- Необязательный пролог
- Тело документа
- Необязательный эпилог, следующий за деревом элементов

Рассмотрим каждую из частей более подробно.

Пролог

Пролог состоит из нескольких частей:

Необязательное *объявление XML* (XMLDeclaration).

Объявление заключено между символами `<?...?>` и содержит: пометку `xml` и номер версии (*version*) спецификации XML; указание на кодировку символов (*encoding*), в которой написан

документ (по умолчанию `encoding="UTF-8"`); параметр `standalone`, который может принимать значения "yes" или "no" (по умолчанию `standalone="yes"`). Значение "yes" показывает, что в документе содержатся все требуемые декларации элементов, а "no" - что нужны внешние определения DTD.

Все это вместе может выглядеть следующим образом:

```
<?xml version="1.0" encoding="windows-1251" standalone="yes"?>
```

Важно отметить, что в объявлении XML только атрибут *version* является обязательным, все остальные атрибуты могут быть опущены и, следовательно, принимать значения по умолчанию. Также нужно помнить, что все эти атрибуты следует указывать только в приведенном выше порядке.

Комментарии

Инструкции по обработке.

Назначение инструкций по обработке — сообщить информацию, передаваемую XML-процессором приложению. Инструкция по обработке имеет следующую общую форму записи:

```
<? Кому инструкция ?>
```

Здесь *Кому* есть имя приложения, которому адресована инструкция. Допускается любое имя при соблюдении следующих правил:

имя должно начинаться с буквы или символа подчеркивания (`_`), после чего могут следовать или не следовать другие буквы, цифры, точки (`.`), тире (`-`) или символы подчеркивания (`_`); имя «`xml`», в любом сочетании строчных или прописных букв, зарезервировано («`xml`» строчными буквами используется в объявлении XML-документа, которое представляет собой разновидность инструкции по обработке).

Инструкция есть информация, передаваемая приложению. Она может состоять из любой последовательности символов, за исключением пары `?>`, зарезервированной для обозначения окончания инструкции по обработке.

Например, следующая инструкция по обработке предписывает браузеру использовать CSS-таблицу из файла `styles.css`:

```
<?xml-stylesheet type="text/css" href=" styles.css"?>
```

Символы пустых пространств.

Необязательное *объявление типа документа*, DTD (DocumentTypeDeclaration), которое заключено между символами `<!DOCTYPE...>` и может занимать несколько строк. В этой части объявляются теги, использованные в документе, или приводится ссылка на файл, в котором записаны такие объявления.

После объявления типа документа также могут следовать комментарии, команды обработки и символы пустых пространств.

Поскольку все эти части необязательны, пролог может быть опущен.

Тело документа

Тело документа состоит из одного или больше *элементов*. В правильно оформленном XML-документе элементы формируют простое иерархическое дерево, в котором обязательно присутствует корневой элемент (*rootelement*), в который вложены все остальные элементы документа. Имена элементов должны быть уникальны в пределах

документа. Имя корневого элемента считается именем всего документа и указывается во второй части пролога после слова Doctype.

Элемент начинается открывающим тегом, затем идет необязательное содержимое элемента, после чего записывается закрывающий тег (в отличие от HTML, наличие закрывающего тега обязательно, исключением являются элементы без содержания, так называемые *пустые элементы*, которые могут быть записаны в сокращенной форме: <имя_элемента/>). В качестве содержимого элемента могут выступать:

Другие элементы

Символьные данные

Ссылки на символы

Для того чтобы вставить в текст документа некоторый символ, который, например не присутствует в раскладке клавиатуры либо может быть неправильно истолкован анализатором, используют ссылки на символы. Ссылка на символ обязательно начинается со знака "&" (амперсанта) и заканчивается точкой с запятой. Ссылки на символы записываются в следующем виде:

&# код_символа_в_Unicode;

Код символа можно записать и в шестнадцатеричном виде. В этом случае перед ним ставится символ "x":

&#xШестнадцатеричный_код_символа;

Ссылки на сущности

Ссылки на сущности позволяют включать любые строковые константы в содержание элементов или значение атрибутов. Ссылки на сущности, как и ссылки на символы, начинающиеся с амперсанта, после которого идет имя сущности и заканчивающиеся точкой с запятой:

&имя_сущности;

Ссылки на сущности указывают программе-анализатору подставить вместо них строку символов, заранее заданную в определении типа документа (DTD).

Комментарии

Если надо вставить в текст документа комментарий либо сделать какой-то фрагмент "невидимым" для программы-анализатора, то его оформляют следующим образом:

<!--...текст комментария...-->

Разделы **CDATA**

Секция CDATA используется, для того чтобы задать область документа, которую при разборе анализатор будет рассматривать как простой текст, игнорируя любые инструкции и специальные символы. Программа-анализатор не разбивает секцию CDATA на элементы, а считает ее просто набором символов. В отличие от комментариев, содержание данной секции не игнорируется, а передается без изменений на выход программы анализатора, благодаря чему его можно использовать в приложении.

Секция CDATA начинается со строки <![CDATA[, после которой записывается содержимое секции. Завершается секция двумя закрывающими квадратными скобками и знаком "меньше":

<![CDATA[содержание секции]]>

Инструкции по обработке

Инструкции по обработке содержат указания программе-анализатору документа XML. Инструкции по обработке заключаются между символами <? и ?>. Сразу за

начальным вопросительным знаком записывается имя программного модуля, которому предназначена инструкция. Затем, через пробел, идет сама инструкция, передаваемая программному модулю. Сама инструкция это обычная строка, которая не должна содержать набор символов ">", означающий конец инструкции. Примером инструкции по обработке может служить строка объявления XML:

```
<?xmlversion="1.0" encoding="windows-1251"?>
```

Эта инструкция предназначена программе, обрабатывающей документ XML. Инструкция передает ей номер версии и кодировку, в которой записан документ.

Атрибуты

Открывающие теги либо теги пустых элементов в XML могут содержать атрибуты, представляющие собой пару *имя=значение*. В одном открывающем теге разрешается использовать только один экземпляр имени атрибута. Атрибуты могут содержать ссылки на объекты, ссылки на символы, текстовые символы. В отличие от языка HTML, в XML значения атрибутов обязательно надо заключать в апострофы ('), либо в кавычки (").

Таким образом, атрибут может быть записан в одном из двух форматов:

```
имя_атрибута="значение_атрибута" имя_атрибута='значение_атрибута'
```

Атрибуты используются для того, чтобы связать некоторую информацию с элементом, а не просто включить ее в содержание последнего. Однозначного ответа на вопрос «Что лучше выбрать – элемент или атрибут?» не существует. Каждый выбирает то, что ему больше нравится. Атрибуты удобно использовать для описания простых значений или для указания типа элемента. Например, мы можем ввести в открывающий тег <city> атрибут type (который может принимать одно из значений: город, поселок, деревня). Тогда данный тег может выглядеть следующим образом:

```
<citytype="город"> Новосибирск </city>
```

Эпилог

В эпилог XML могут входить комментарии, инструкции по обработке и/или пустое пространство.

В языке XML не определен индикатор конца документа, большинство приложений для этой цели используют завершающий тег корневого элемента документа. Таким образом, встретив завершающий тег корневого элемента, скорее всего, приложение закончит обработку документа, и эпилог обработан не будет.

Практическая работа № 25. Создание XML документа.

Цель: Изучение возможностей и назначение языка XML. Получение навыков использования XML-технологии для организации хранения информации. Изучение способов перевода текстовых и табличных данных в формат XML.

Пример XML-документа

Ниже представлена структура XML-документа по приведенной статической информационной модели предметной области «Расписание занятий».

```
<?xml version="1.0"?>
<timetable>
  <day dayOfWeek="Monday">
    <lesson type="practical">
      <timeFrom>08.00</timeFrom>
      <timeTo>09.30</timeTo>
      <subject>Deutsch</subject>
      <teacher>Borisova</teacher>
      <room>216</room>
    </lesson>
    <lesson type="lecture">
      <timeFrom>09.40</timeFrom>
      <timeTo>11.10</timeTo>
      <subject>SAP Administration</subject>
      <teacher>Egorov</teacher>
      <room>384</room>
    </lesson>
    <lesson type="practical">
      <timeFrom>11.20</timeFrom>
      <timeTo>12.50</timeTo>
      <subject>SAP Administration</subject>
      <teacher>Petrov</teacher>
      <room>384</room>
    </lesson>
  </day>
</timetable>
```

Прежде чем Internet Explorer отобразит XML-документ, его встроенный синтаксический XML-анализатор (parser) просмотрит содержимое документа. Если он обнаружит ошибку, Internet Explorer отобразит страницу с сообщением об ошибке, не предпринимая попытки отобразить документ.

Internet Explorer использует для отображения документа имеющуюся по умолчанию таблицу стилей. Именно поэтому в описании ошибки упоминается использование таблицы стилей XSL.

Практическая часть

Задание для самостоятельного выполнения

1. Представьте в виде древовидной структуры данные выбранной по желанию предметной области. Примерами предметных областей могут служить, например, «Издательство» (выпускающее газеты и журналы), «Научная конференция», «Кафедра вуза» или «Составление букетов».

2. Создайте XML-документ, используя информационную модель, построенную в первой части.

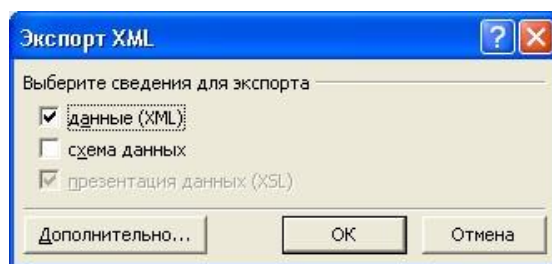
Преобразование в формат XML Рассмотрим процедуру преобразования объекта базы данных в формат XML:

1. В окне базы данных выделите необходимый объект щелчком левой кнопки мыши и выберите команду **Файл, Экспорт** (File, Export) или щелкните по объекту правой кнопкой и выберите в контекстном меню команду **Экспорт** (Export).

2. Появится диалоговое окно **Экспорт объекта** (Export To). В раскрывающемся списке **Типы** (Save as type) выделите элемент **Документы XML** (XML Documents). В поле **Имя файла** (File name) введите название документа HTML и нажмите кнопку **Экспорт** (Export).

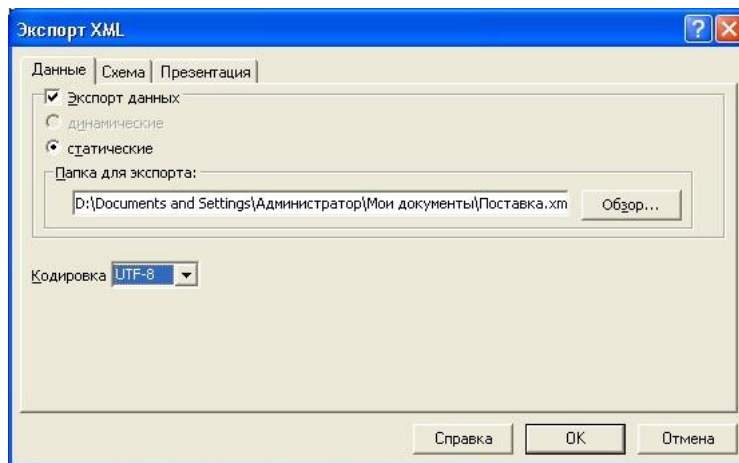
3. Появится диалоговое окно **Экспорт XML** (Export XML).

4. Это диалоговое окно содержит три флажка для трех составляющих документа XML: данных, схемы данных и представления данных. Установите флажки для тех частей документа XML, которые необходимо получить в результате экспорта. Например, если установить одновременно флажки **данные (XML)** (Data (XML)) и **схема данных** (Schema of the data), то схема данных будет экспортирована и сохранена отдельно от файла данных XML, в файле XSD. Для таблиц можно экспортировать только данные или их структуру или данные и структуру вместе и не экспортировать представление таблицы в виде Web-страницы: флажок **презентация данных (XSL)** (Presentation of your data (XSL)) можно снять.



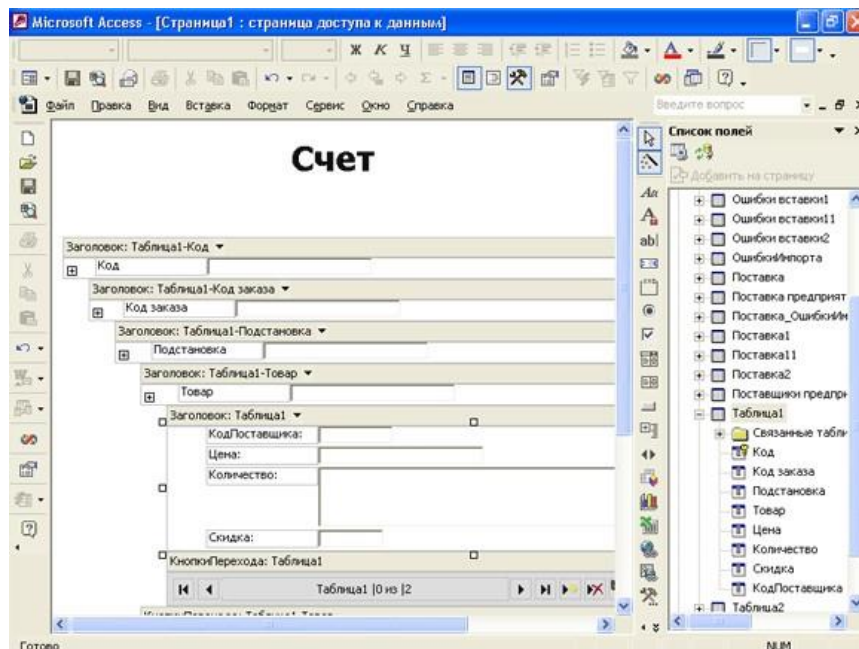
Флажок **данные (XML)** следует сбросить лишь в том случае, если вы хотите экспортировать только новое представление объекта базы данных, а данные были экспортированы в файл XML раньше. Иначе при открытии полученной Web-страницы будет выдано сообщение об ошибке подключения к источнику данных, а сама страница окажется практически пустой.

5. Чтобы установить дополнительные параметры преобразования, нажмите кнопку **Дополнительно** (Advanced).

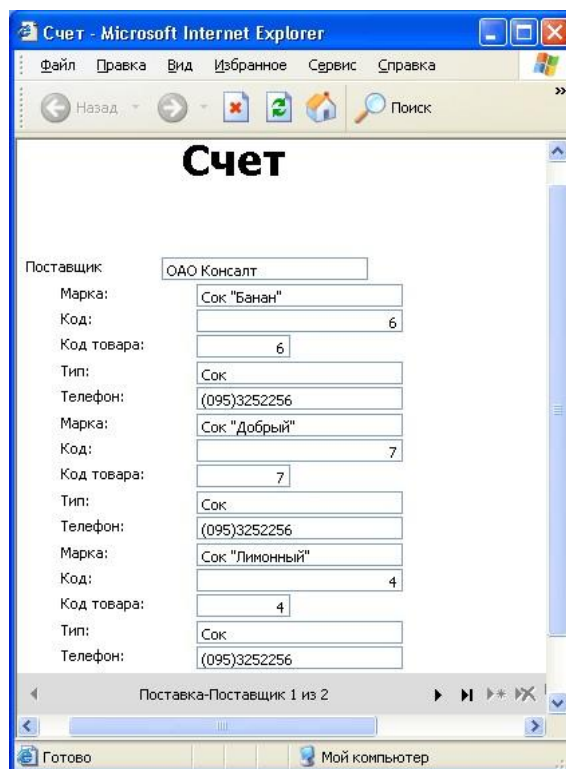


6. Кроме того, это диалоговое окно содержит еще две вкладки для двух оставшихся составляющих документа XML: схемы данных и представления данных. На каждой из этих вкладок можно установить флажок, позволяющий экспортировать соответствующую часть документа XML, указать путь и имя файла, в котором будет сохранена эта часть документа, и задать некоторые дополнительные параметры:

- на вкладке **Данные** (Data) с помощью раскрывающегося списка **Кодировка** (Encoding) можно выбрать кодировку для сохранения содержимого страницы;
- на вкладке **Схема** (Schema) с помощью переключателей можно выбрать, экспортировать ли структуру данных в отдельный файл XSD или внедрить ее в файл XML вместе с данными;



- на вкладке **Презентация** (Presentation) с помощью переключателей можно выбрать формат представления Web-страницы: обрабатываемый на стороне клиента HTML или на стороне сервера ASP. Если экспортируемый объект содержит рисунки, которые необходимо включить в результирующий документ, можно указать имя папки, в которой они расположены. На этой же вкладке есть поле для ввода имени файла XSL с описанием оформления страницы. Если указанный файл не существует, Access создаст файл в формате XSL описанием внешнего представления документа (разработчик может использовать его в дальнейшем как основу для разработки файла в формате XSLT для преобразования документа XML в другой формат).



7. Нажмите кнопку ОК.

8. Если предложенные параметры преобразования не были изменены, в результате будут созданы три файла: XML — с данными, XSL — со стилями и HTM — с представлением преобразованного объекта. Чтобы открыть полученную Web-страницу, требуется открыть в Internet Explorer файл HTM или ASP, в зависимости от того, какой формат представления был выбран..

К сожалению, подчиненные формы и отчеты, а также связанные рисунки не будут добавлены в документ XML, полученный из объекта базы данных. Однако простые объекты с внедренными рисунками, экспортированные в формат XML, будут выглядеть практически точно так же, как исходные в Access. Если данных в источнике данных выбранного объекта много, загрузка Web-страницы в программу просмотра Интрнета может занять существенное время.

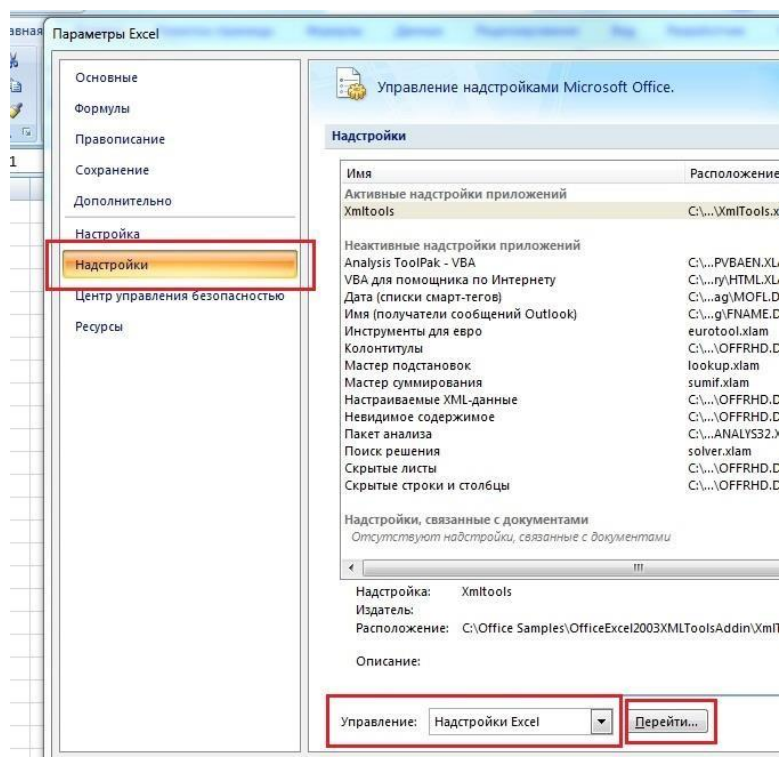
Поскольку сами данные, содержащиеся в таком документе Web, полностью хранятся в файле XML, нет нужды в доступе к базе данных, из которой изначально были взяты эти данные (как, например, и в случае преобразования отчета в формат XML). Это означает, что для обеспечения доступа к такому документу достаточно поместить только упомянутый набор файлов, содержащих описание этого документа, на сервер Web. С другой стороны, это означает, что изменение данных в таком документе возможно лишь с помощью приложений, имеющих доступ к файлу XML и способных воспринять этот формат.

Перевод XLS-таблицы в XML-формат 1)

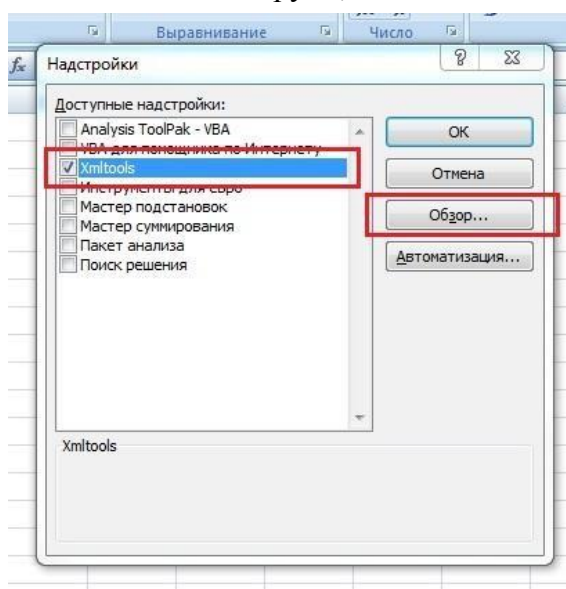
Загрузите надстройку для работы с XML.

2) Заходим в главное меню Excel, параметры.

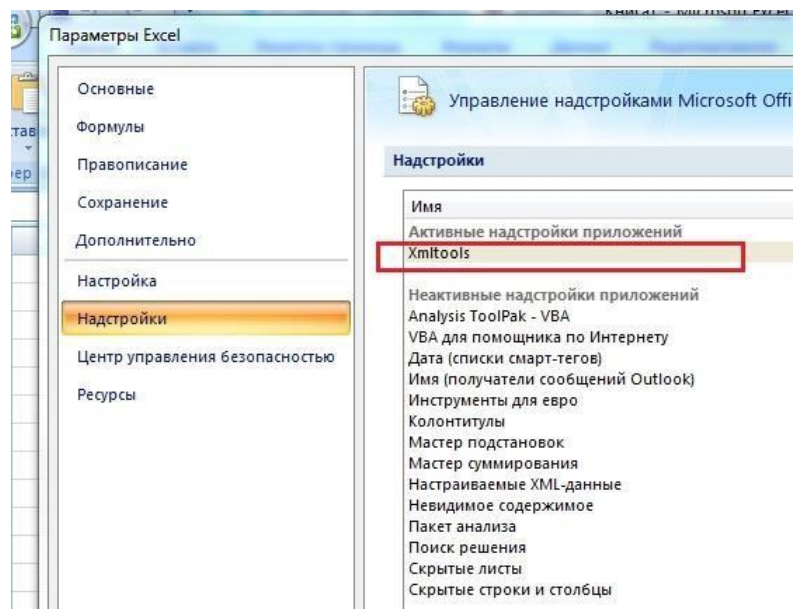
3) В открывшемся меню выбираем «Надстройка». Ищем пункт «Управление», выбираем «Надстройки Excel» и нажимаем «Перейти»:



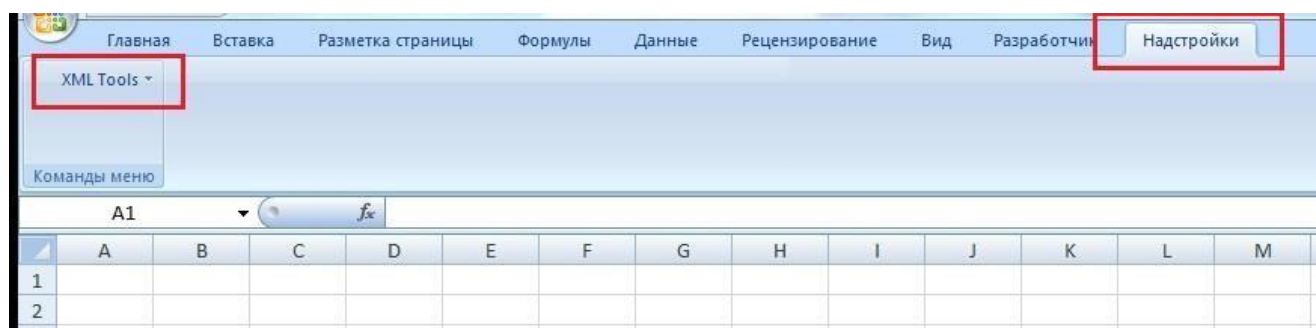
4) В открывшемся окне через «Обзор» выбираем нужный нам файл **XmlTools.xla**, добавляем его в панель и активируем, поставив галочку напротив него.



5) Если вы все сделали правильно, то в «Параметрах» в разделе «Надстройки» вы увидите установленное расширение:

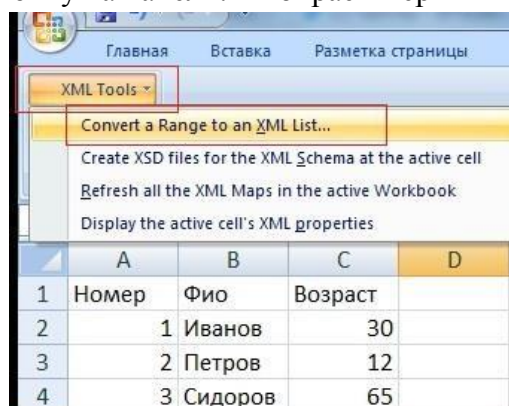


Далее, все просто. У вас в меню появляется новая вкладка «Надстройки», а на ней заветная кнопка «XML Tools»:

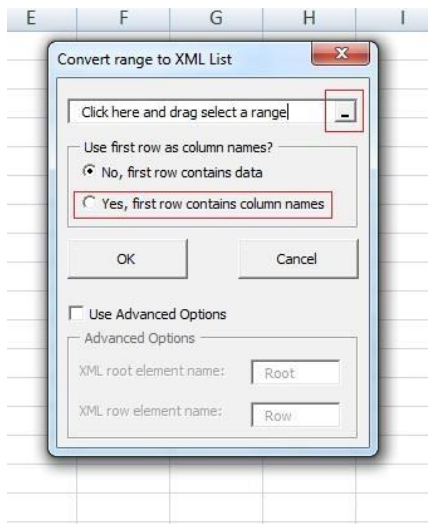


Теперь берем наши данные для создания XML документа и работаем с ними.

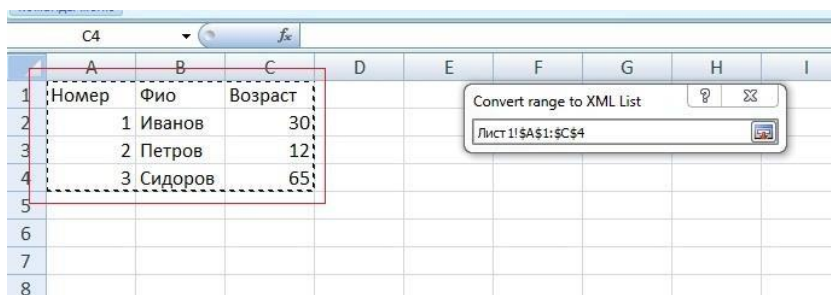
6) Нажимаем на кнопку на панели. Выбираем первый пункт:



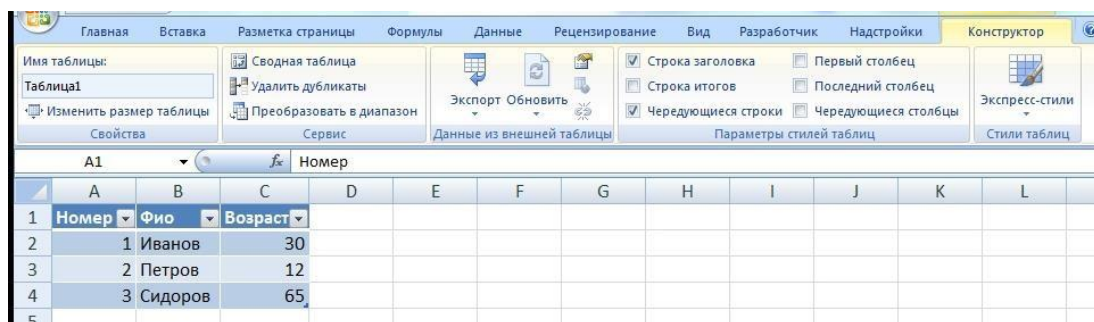
7) В появившемся окне выбираем yes/no если наши данные содержат / не содержат заголовка, в моем случае содержат, потому переставляю галочку на yes. Здесь же зажимаем маленькую «пипку», чтобы выбрать ячейки с данными:



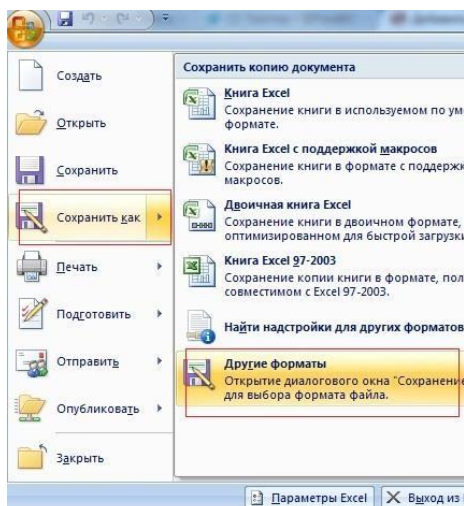
8) Протягиванием выбираем данные для создания XML — документа. Отжимаем кнопку и жмем ОК.



Если все сделали верно, то табличка станет сине-белой, и меню изменится:



Ну, уже почти все. Теперь идем в параметры сохранения документа и там выбираем XML — данные.



Мы получили на выходе готовый XML-документ.

Контрольные вопросы:

1. Язык разметки XML? Достоинства XML?
2. Что представляют собой XML-документы?
3. Что такое информационная модель? Статическая информационная модель?
4. Дайте определения понятиям: Модели рабочих процессов, Модели потоков данных, Объектные модели, Жизненные циклы объекта.
5. Опишите синтаксис XML

Практическая работа № 26.

Проведение стоимостного анализа с помощью Ramus educational.

Цель: Научиться производить стоимостный анализ, проводить расчёт выполнения работ на разных стадиях бизнес-процесса.

В результате освоения учебной дисциплины обучающийся должен **уметь:**

- ☐ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;
- ☐ использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения.

В результате освоения учебной дисциплины обучающийся должен **знать:**

- ☐ требования к проектируемой системе;
- ☐ модели жизненного цикла информационной системы, методы проектирования информационной системы;
- ☐ технологии проектирования информационной системы, оценку и управление качеством информационной системы.

Теоретическая часть

Стоимостный анализ и свойства, определяемые пользователем

Обычно сначала строится *функциональная модель* существующей организации работы - **AS-IS (как есть)**. После построения модели AS-IS проводится анализ бизнес-процессов, потоки данных и объектов перенаправляются и улучшаются, в результате строится модель **ТО-ВЕ (как должно быть)**. Как правило, строится несколько моделей ТО-ВЕ, из которых по какому-либо критерию выбирается наилучшая. Проблема состоит в том, что таких критериев много и непросто определить важнейший. Для того чтобы определить качество созданной модели с точки зрения эффективности бизнес-процессов, необходима система метрики, т. е. качество следует оценивать количественно.

Ramuseducational предоставляет аналитику два инструмента для оценки модели – **стоимостный анализ, основанный на работах (Activity Based Costing, ABC)**, и **свойства, определяемые пользователем (User Defined Properties, UDP)**. ABC является широко распространенной методикой, используемой международными корпорациями и государственными организациями для идентификации истинных движителей затрат в организации.

Стоимостный анализ представляет собой соглашение об учете, используемое для сбора затрат, связанных с работами, с целью определить общую стоимость процесса. Стоимостный анализ основан на модели работ, потому что количественная оценка невозможна без детального понимания функциональности предприятия. Обычно ABC применяется для того, чтобы понять происхождение выходных затрат и облегчить выбор нужной модели работ при реорганизации деятельности предприятия (Business Process Reengineering, BPR). С помощью стоимостного анализа можно решить такие задачи, как определение действительной стоимости производства продукта, определение действительной стоимости поддержки клиента, идентификация работ, которые стоят больше всего (те, которые должны быть улучшены в первую очередь), обеспечение менеджеров финансовой мерой предлагаемых изменений т. д.

ABC может проводиться только тогда, когда модель работы последовательная (следует синтаксическим правилам IDEF0), корректная (отражает бизнес), полная (охватывает всю рассматриваемую область) и стабильная (проходит цикл экспертизы без изменений), другими словами, создание модели работы закончено.

ABC включает следующие основные понятия:

- **объект затрат** – причина, по которой работа выполняется, обычно, основной выход работы, стоимость работ есть суммарная стоимость объектов затрат («Готовое изделие»);
- **движитель затрат** – характеристики входов и управлений работы («Сырье», «Чертеж»), которые влияют на то, как выполняется и как долго длится работа;
- **центры затрат**, которые можно трактовать как статьи расхода.

При проведении стоимостного анализа в BPwin сначала задаются единицы измерения времени и денег. Для задания единиц измерения следует вызвать диалог *Model Properties* (меню *Edit/Model Properties*), закладка *ABC Units* (рис. 1).

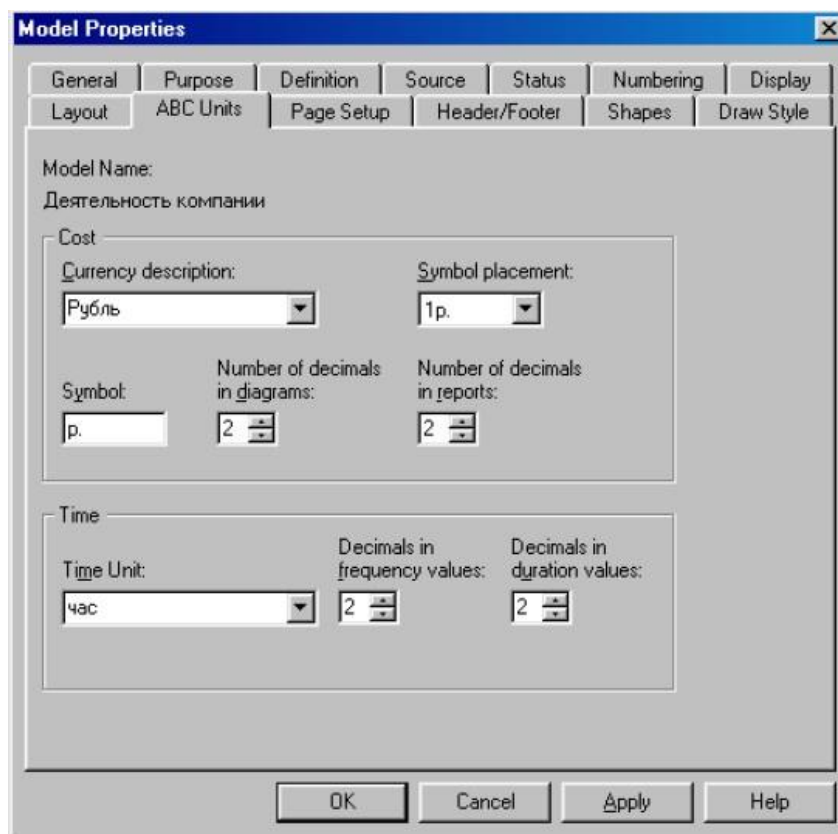


Рис. 1. Настройка единиц измерения валюты и времени

Если в списке выбора отсутствует необходимая валюта (например, рубль), ее можно добавить. Символ валюты по умолчанию берется из настроек Windows. Диапазон измерения времени в списке *Unit of measurment* достаточен для большинства случаев – от секунд до лет.

Затем описываются центры затрат (*cost centers*). Для внесения центров затрат необходимо вызвать диалог *Cost Center Editor* (меню *Model/ABC Cost Centers*).

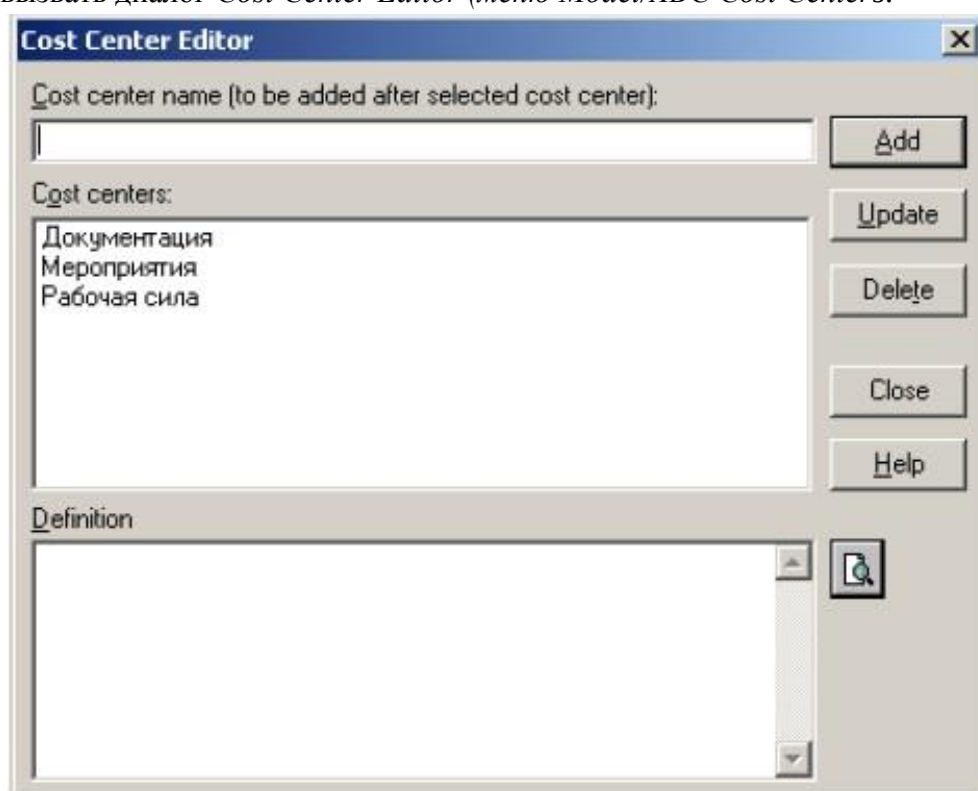


Рис.2 Диалог Cost Center Editor

Каждому центру затрат следует дать подробное описание в окне *Definition*. Список центров затрат упорядочен. Порядок в списке можно менять при помощи стрелок, расположенных справа от списка. Задание определенной последовательности центров затрат в списке, во-первых, облегчает последующую работу при присвоении стоимости работам, а во-вторых, имеет значение при использовании единых стандартных отчетов в разных моделях. Хотя BPwin сохраняет информацию о стандартном отчете в файле BPWINRPT.INI, информация о центрах затрат и UDP сохраняется в виде указателей, т. е. хранятся не названия центров затрат, а их номера. Поэтому, если нужно использовать один и тот же стандартный отчет в разных моделях, списки центров затрат должны быть в них одинаковы.

Для задания стоимости работы (для каждой работы на диаграмме декомпозиции) следует щелкнуть правой кнопкой мыши по работе и на всплывающем меню выбрать *Cost*. В диалоге *Activity Cost* указывается частота проведения данной работы в рамках общего процесса (окно *Frequency*) и продолжительность (*Duration*). Затем следует выбрать в списке один из центров затрат и в окне *Cost* задать его стоимость. Аналогично назначаются суммы по каждому центру затрат, т. е. задается стоимость каждой работы по каждой статье расхода. Если в процессе назначения стоимости возникает необходимость внесения дополнительных центров затрат, диалог *Cost Center Editor* вызывается прямо из диалога *Activity Cost* соответствующей кнопкой.

Cost Center	рубль
Документация	2 700,00
Мероприятия	41 800,00
Рабочая сила	28 500,00

Data is from Decompositions. Total cost: 73 000,00
☐ Override decompositions Total cost x Frequency: 73 000,00
☒ Compute from decompositions

Frequency: 1,00
Duration: 160,00 час
Duration x Frequency 160,00 час

Cost Center Editor...

OK Отмена Применить Справка

Рис. 3 Задание стоимости работ в диалоге Activity Cost

Общие затраты по работе рассчитываются как сумма по всем центрам затрат. При вычислении затрат вышестоящей (родительской) работы сначала вычисляется произведение затрат дочерней работы на частоту работы (число раз, которое работа выполняется в рамках проведения родительской работы), затем результаты складываются. Если во всех работах модели включен режим *Compute from Decompositions*, подобные вычисления автоматически проводятся по всей иерархии работ снизу вверх.

Этот достаточно упрощенный принцип подсчета справедлив, если работы выполняются последовательно. Встроенные возможности BPwin позволяют разрабатывать упрощенные модели стоимости, которые тем не менее оказываются чрезвычайно полезными при предварительной оценке затрат.

Для проведения более тонкого анализа можно воспользоваться специализированным средством стоимостного анализа *EasyABC* (ABC Technology, Inc.). BPwin имеет двунаправленный интерфейс с *EasyABC*. Для экспорта данных в *EasyABC* следует выбрать пункт меню *File/Export/Node Tree*, задать в диалоге *Export Node Tree* необходимые настройки и экспортировать дерево узлов в текстовый файл (.txt). Файл экспорта можно импортировать в *EasyABC*. После проведения необходимых расчетов результирующие данные можно импортировать из *EasyABC* в BPwin. Для импорта нужно выбрать меню *File/Import/Costs* и в диалоге *Import Activity Costs* выбрать необходимые установки.

Результаты стоимостного анализа наглядно представляются на специальном отчете BPwin – *Activity Cost Report* (меню *Report/Activity Cost Report*). Отчет позволяет документировать имя, номер, определение и стоимость работ, как суммарную, так и раздельно по центрам затрат (рис. 4).



Рис. 4 Диалог настройки отчета по стоимости работ

Результаты отображаются и непосредственно на диаграммах. В левом нижнем углу прямоугольника работы может показываться либо стоимость (по умолчанию), либо продолжительность, либо частота проведения работы.

Настройка отображения осуществляется в диалоге *Model Properties* (меню *Edit/Model Properties*), закладка *Display, ABC Data, ABC Units*.

ABC позволяет оценить стоимостные и временные характеристики системы. Если стоимостных показателей недостаточно, имеется возможность внесения собственных метрик – свойств, определенных пользователем (User Defined Properties, UDP). UDP позволяют провести дополнительный анализ, хотя и без суммирующих подсчетов.

Для описания UDP служит диалог *User-Defined Property Name Editor* (меню *Edit/UDP Definition*) (рис. 5). В верхнем окне диалога вносится имя UDP, в списке выбора *Datatype* описывается тип свойства. Имеется возможность задания 18 различных типов UDP, в том числе управляющих команд и массивов, объединенных по категориям. Для внесения категории следует задать имя категории в окне *New Category/Member* и щелкнуть по кнопке *Add Category*. Для присвоения свойства категории необходимо выбрать UDP из списка, затем категорию из списка категорий и щелкнуть по кнопке *Update*. Одна категория может объединять несколько свойств, в то же время одно свойство может входить в несколько категорий. Свойство типа *List* может содержать массив предварительно определенных значений.

Для определения области значений UDP типа *List* следует задать значение свойства в окне *New Category/Member* и щелкнуть по кнопке *Add Member*.

Значения из списка можно редактировать и удалять.

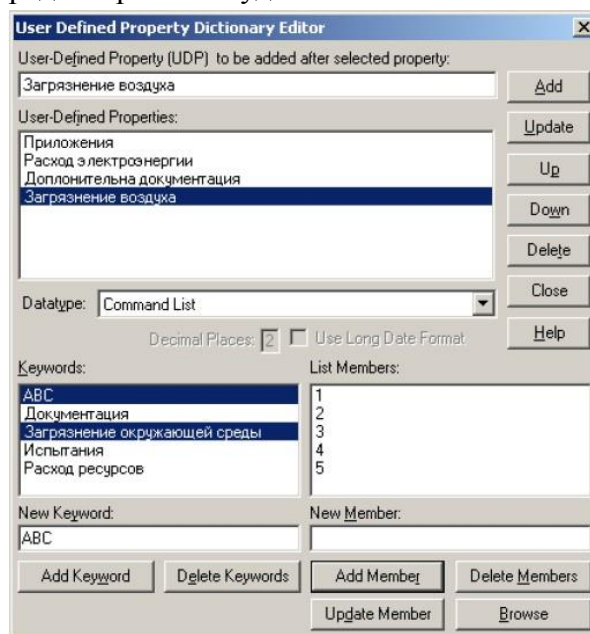


Рис. 5 Диалог описания UDP

Каждой работе можно поставить в соответствие набор UDP. Для этого следует щелкнуть правой кнопкой мыши по работе и выбрать пункт меню *UDP Editor*. В закладке *UDP Values* диалога *IDEF0 Activity Properties* можно задать значения UDP. Свойства типа *List* отображаются списком выбора, который заполнен предварительно определенными значениями. Свойства типа *Command* могут иметь в качестве значения командную строку.

Кнопка *Categories* служит для задания фильтра по категориям UDP. По умолчанию в списке показываются свойства всех категорий.

Результат задания проанализировать в отчете *Diagram Object Report* (меню *Report/Diagram Object Report*) (рис. 6).

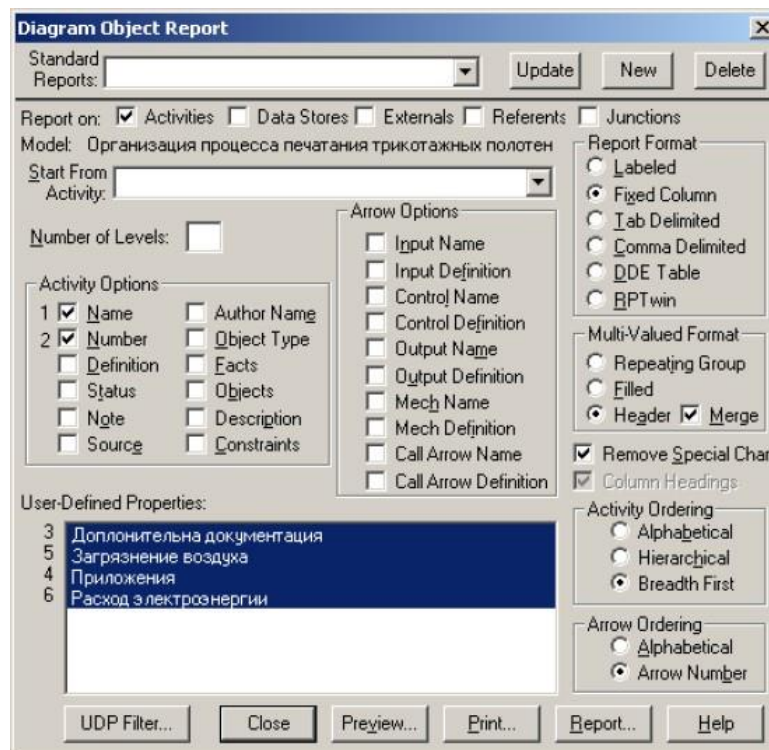


Рис. 6 Диалог настройки отчета Diagram Object Report **Практическая работа № 27.**

Работа в системе Ramus educational.

Цель: Научиться производить стоимостный анализ, проводить расчёт выполнения работ на разных стадиях бизнес-процесса.

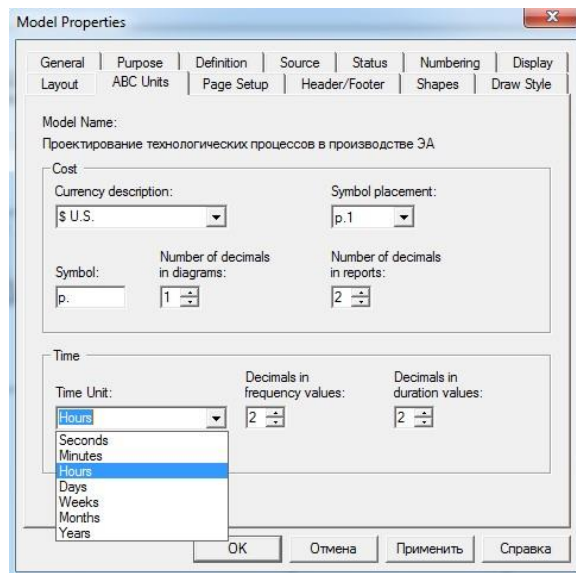
Практическая часть

Задание: Произвести стоимостный анализ построенных в предыдущих практических работах моделей.

Пример проведения стоимостного анализа

В диалоговом окне **ModelProperties** (вызывается из меню **Model/Modelproperties**) во вкладке **ABCUnits** установите единицы измерения денег – рубли и времени часы.

AllFusion Process Modeler r7 - [Cost Center Dictionary]	
Dictionary Edit View Help	
Name	Definition
Documentacia	Zatrady na razrabotku dokumentacii
Meropriyatia	Zatrady na process proectirovaniia
Raboचाia sila	Zatrady na zarplaty personala



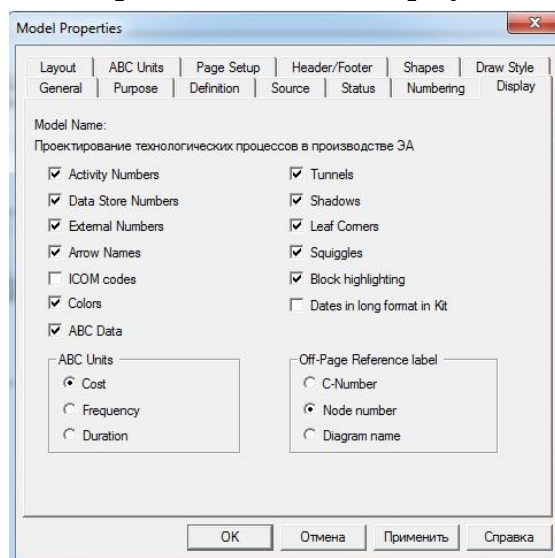
Перейдите в меню **Dictionary/CostCenter (Словарь/Центр затрат)** и в окне **CostCenterDictionary (Словарь Центра Затрат)** внесите название и определение центров затрат.

Таблица 1 - Центры затрат ABC

Центр затрат	Определение
Документация	Затраты на разработку документации
Мероприятия	Затраты на процесс проектирования
Рабочая сила	Затраты на зарплату персоналу

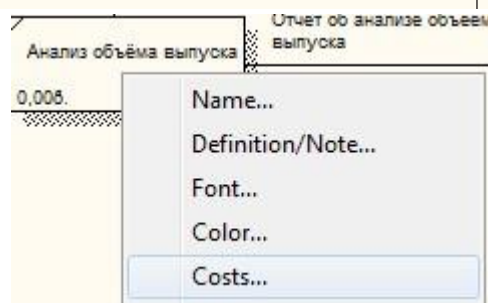
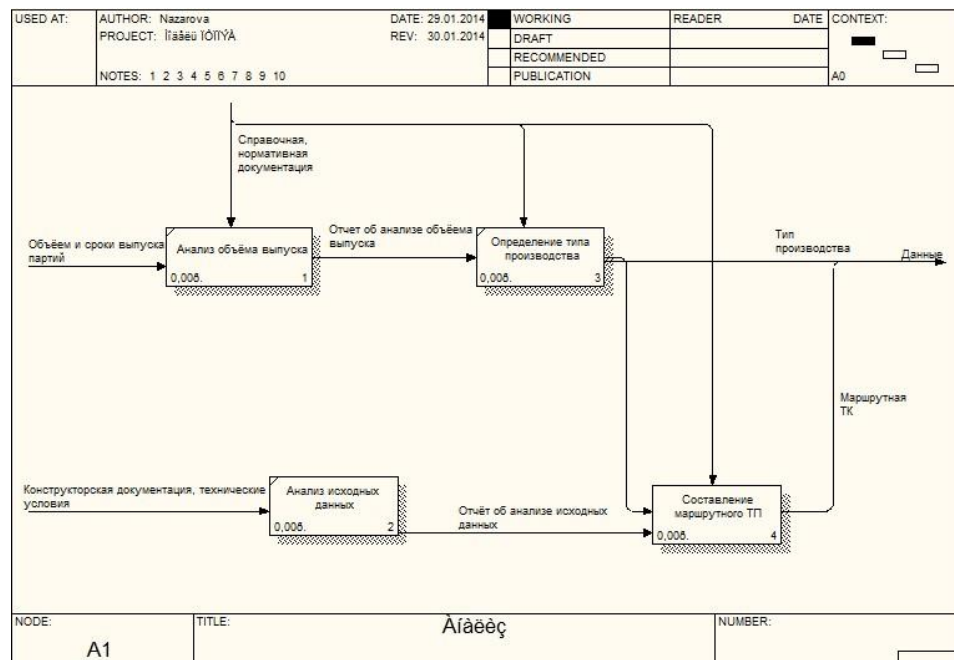
Видокна **Cost Center Dictionary** следующий.

Для отображения стоимости **КАЖДОЙ РАБОТЫ** в нижнем углу прямоугольника перейдите в меню **Model/Model Properties** и вкладки **Display** включите опцию **ABC Data**.



Для отображения частоты или продолжительности работы переключите радиокнопки в группе **ABCUnits**.

Для назначения стоимости работе (например, «Анализ») на диаграмме **A0** следует открыть диаграмму **A1**, щелкнуть по работе (например, «Анализ объёма выпуска») правой кнопкой мыши и выбрать в контекстном меню **Cost**.



Откроется диалоговое окно **ActivityProperties**, в котором следует указать величины затрат (в рублях) на компоненты, рабочую силу, управление и временные характеристики работы – **Duration (Продолжительность)** и **Frequency (Частоту)** выполнения.

Activity Properties
X

UDP Values	UOW	Source	Roles	Box Style
Name	Definition	Status	Font	Color

Activity Name:
 Анализ объема выпуска

Cost Center	\$ U.S.
Documentacia	25,00
Meropriyatia	36,00
Rabochaia sila	42,00

This Activity has NO Decomposition.
 Total cost: 103,00

☐ Override decompositions
 Total cost x Frequency: 103,00

☐ Compute from decompositions
 Cost Center Editor...

Frequency:

Duration: Minutes

Duration x Frequency 0,000 Minutes

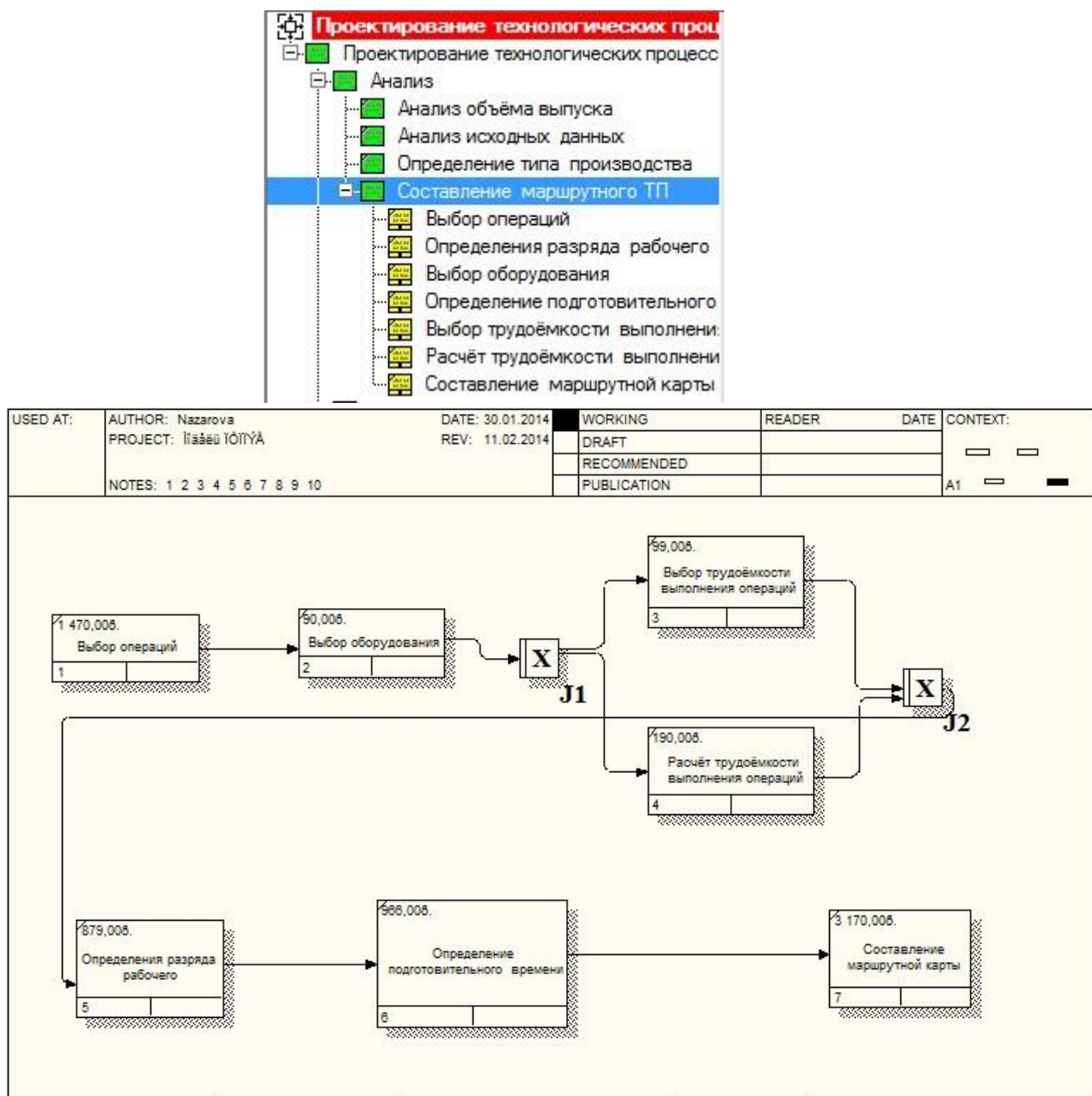
OK
Отмена
Применить
Справка

Далее для **КАЖДОЙ РАБОТЫ** на диаграмме **A1** указать приведённые величины.

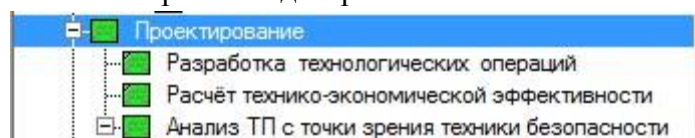
Результат – стоимость работ верхнего уровня отображается в нижнем углу прямоугольника работы.



Для назначения стоимости работы **A1.4** (в данном примере «Составление маршрутного ТП»), необходимо назначить стоимости **КАЖДОЙ РАБОТЫ** на диаграмме **IDEF3 A 14.1**.

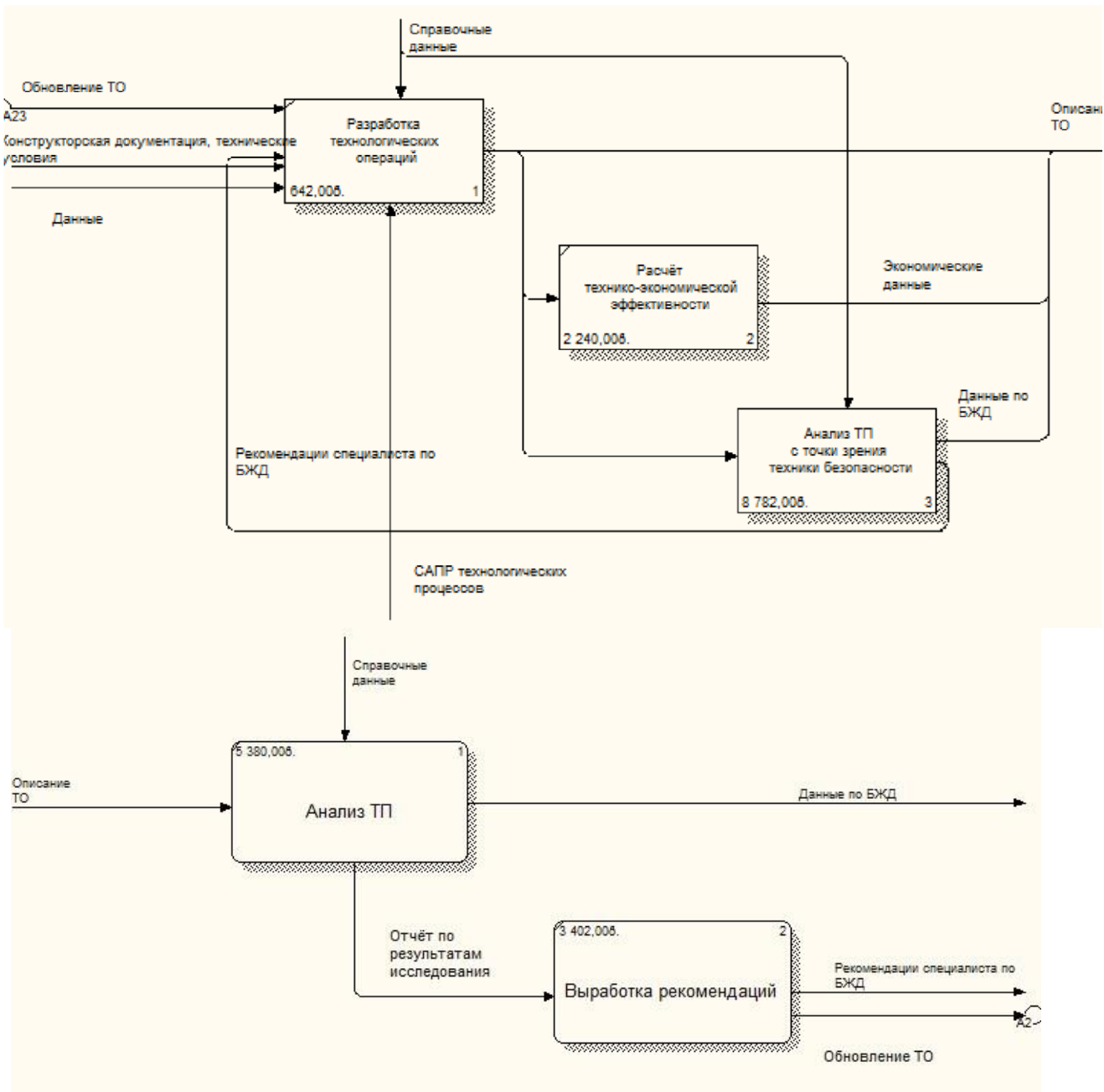
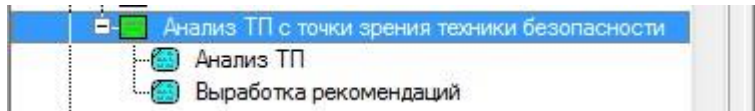


Для назначения стоимости работе «Проектирование» необходимо назначить стоимости работам диаграммы **A2**

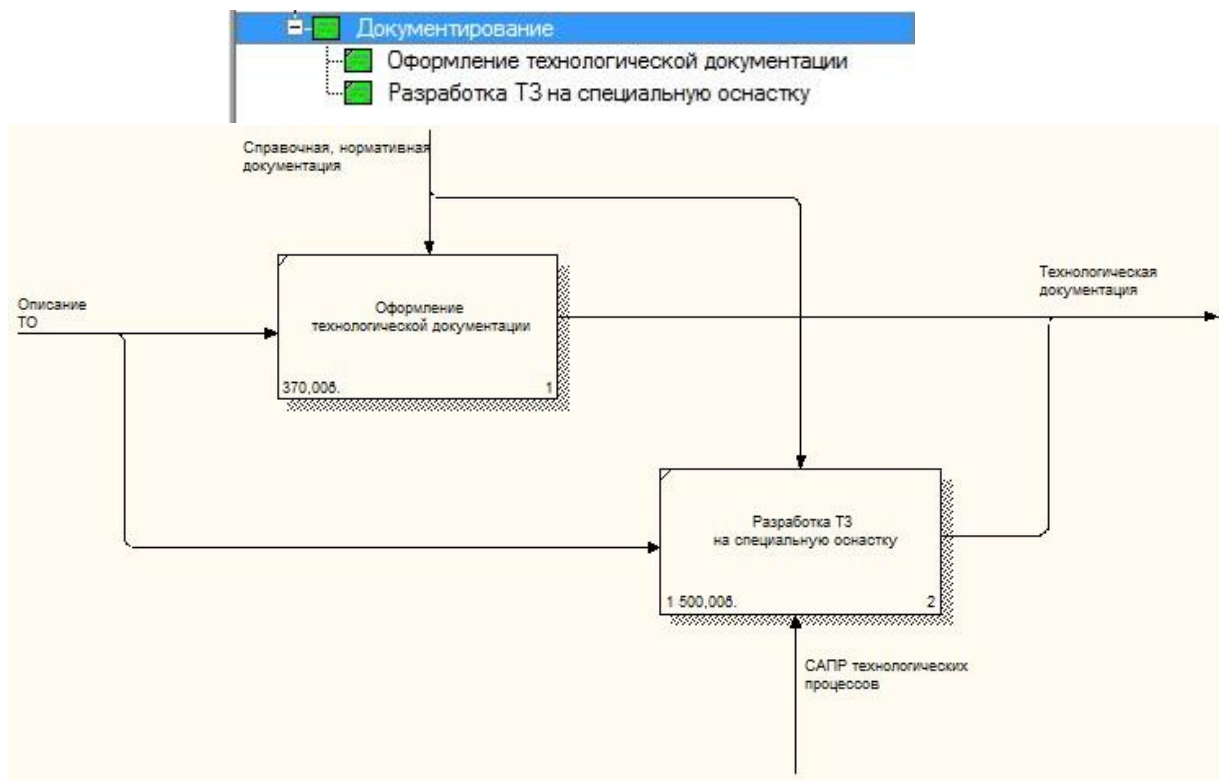


, а для работы «Анализ ТП с

точки зрения техники безопасности», необходимо назначить стоимости работам на DFD диаграмме A2.3



Для назначения стоимости работе «Документирование» необходимо назначить стоимости работам диаграммы А3.



Сгенерируйте отчет **Activity Cost Report (Tools → Activity Cost Report)**.
 В открывшемся диалоговом окне **Activity Cost Report** задайте параметры генерации отчета **Activity Cost Report**.

Costing

Activity Cost Report Preview			
Report Format: Column			
Name	Activity Cost (\$ U.S.)	Cost Center	Cost Center Cost (\$ U.S.)
Проектирование технологических процессов в производстве 3A	15 425,00	Documentacia	3 566,00
		Meropriyatia	2 444,00
		Rabochaia sila	9 415,00
Анализ	1 891,00	Documentacia	676,00
		Meropriyatia	589,00
		Rabochaia sila	626,00
Анализ объёма выпуска	103,00	Documentacia	25,00
		Meropriyatia	36,00
		Rabochaia sila	42,00
Анализ исходных данных	97,00	Documentacia	58,00
		Meropriyatia	23,00
		Rabochaia sila	16,00
Определение типа производства	204,00	Documentacia	58,00
		Meropriyatia	69,00

Практическая работа №28.

AllFusion Process Modeler r7 - [UDP Dictionary]		
Dictionary Edit View Help		
Name	Definition	UDP Datatype
Prilozhenia		Text

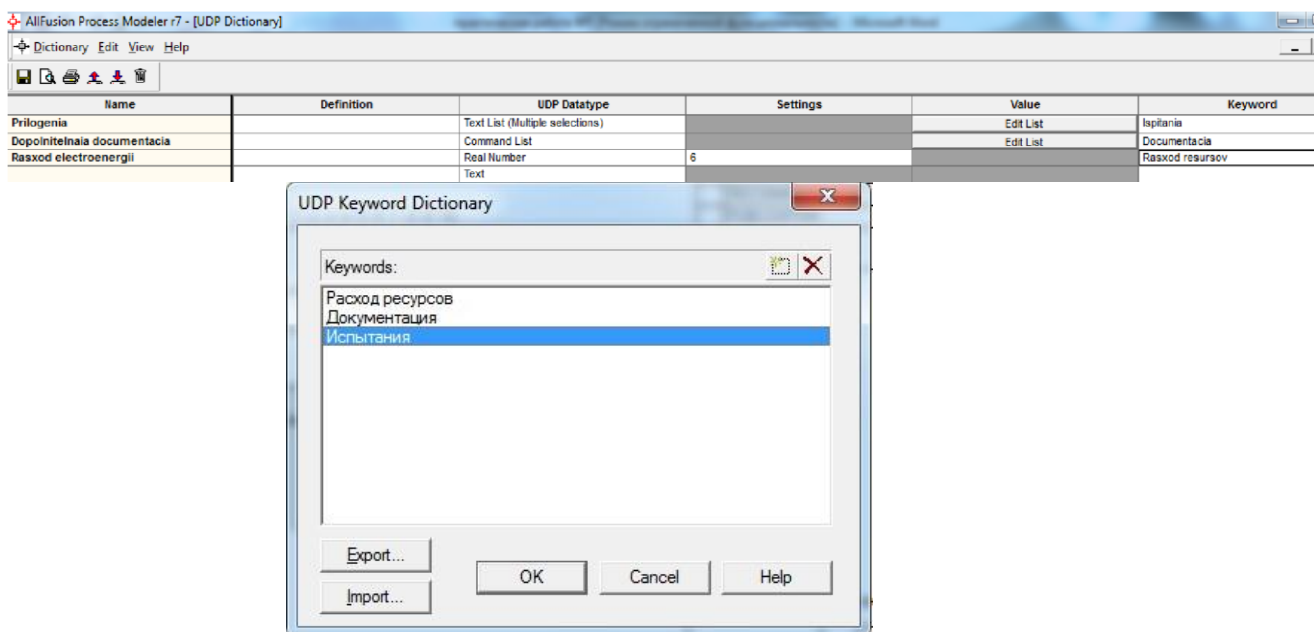
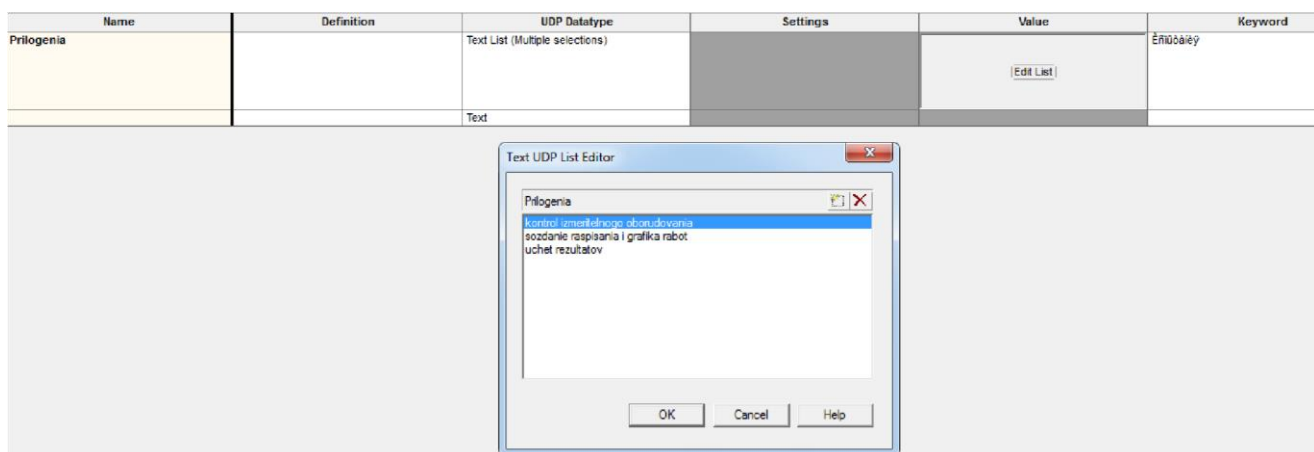
Использование категорий UDP.

Цель: Научиться производить стоимостный анализ, проводить расчёт выполнения работ на разных стадиях бизнес-процесса.

Использование категорий UDP

Перейдите в меню **Dictionary/UDP Keywords** и в диалоговом окне **UDP Keyword Dictionary** внесите ключевые слова UDP (User Defined Properties – Свойства, Определяемые Пользователем).

Например: Расход ресурсов; Документация; Испытания.



Создайте **UDP**. Для этого перейдите в меню **Dictionary/UDP** и в словаре внесите имя **UDP**

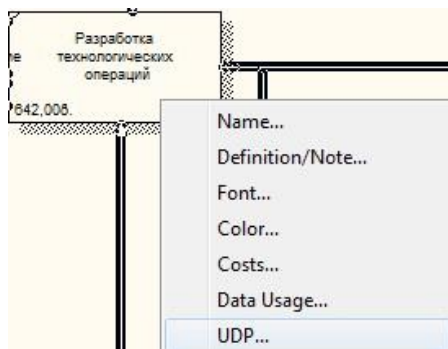
Например «Приложения».

Для **UDP** типа **List (Список)** необходимо в поле **Value** задать список значений.

Для UDP – «Приложения» внесите значения. Например: «Контроль измерительного оборудования, создание расписания и графика работ, учет результатов». В графе **Keyword** указать значение. Например: «Испытания».

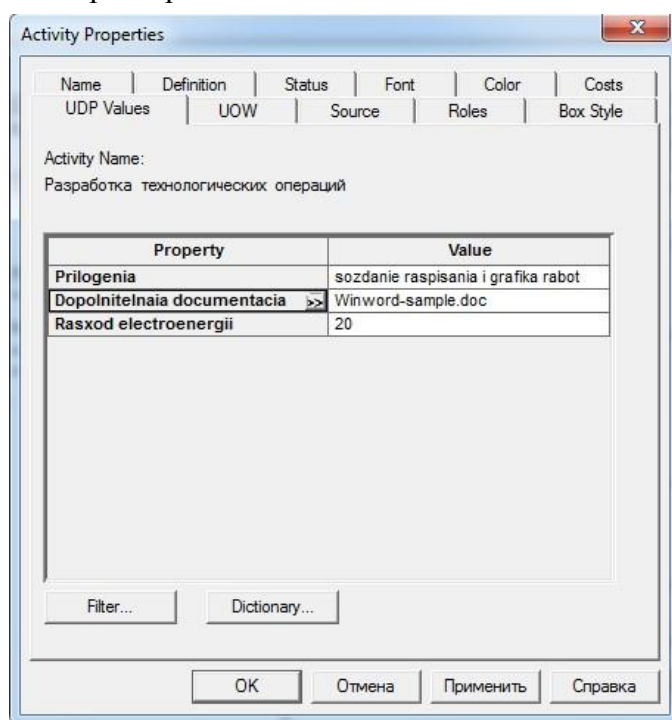
Затем создайте новые UDP и внесите новые значения. Например:

Назначим UDP работе «*Разработка технологических операций*» диаграммы **A2**. Для этого следует щелкнуть по работе правой кнопкой мыши и выбрать в контекстном меню **UDP**. Появится вкладка **UDP Values** диалога **Activity Properties**.



После внесения **UDP** типа **Command** или **CommandList** (см. выше) щелчок по кнопке  приведет к запуску соответствующего приложения.

Примечание— Для того чтобы соответствующее приложение было запущено, необходимо чтобы оно было предварительно создано.



При использовании функции Filter, можно отключить ключевые слова, и в диалоге **ActivityProperties** не будут отображаться UDP с неиспользованными ключевыми словами.

Activity Properties

Name	Definition	Status	Font	Color
UDP Values	UOW	Source	Roles	

Activity Name:
Разработка технологических операций

Property	Value
Prilogenia	sozdanie raspisania i grafi
Dopolnitelnaia documentacia >>	Winword-sample.doc
Rasxod electroenergii	20

Filter... Dictionary...

OK Отмена Применить

Diagram Object UDP Filter

Keywords:

- Documentacia
- Ispitania
- Rasxod resursov

OK Cancel Help

Display UDPs assigned to:

☒ Any
☐ Every selected Keyword.

☒ Display UDPs without Keywords

Select All Clear All

В итоге после фильтрации получим:

Activity Properties

Name	Definition	Status	Font	Color	Costs
UDP Values	UOW	Source	Roles	Box Style	

Activity Name:
Разработка технологических операций

Property	Value
Prilogenia	sozdanie raspisania i grafika robot ▾
Dopolnitelnaia documentacia >>	Winword-sample.doc

Filter... Dictionary...

OK Отмена Применить Справка

Важно! Свойства UDP можно присвоить не только работам, но и стрелкам.

Практическая работа № 29.

Ramus educational основные характеристики.

Цель: Научиться производить стоимостный анализ, проводить расчёт выполнения работ на разных стадиях бизнес-процесса.

Контрольные вопросы:

1. С какой целью проводится стоимостный анализ?
2. Когда можно проводить стоимостный анализ?
3. Каков порядок проведения стоимостного анализа?
4. Насколько глубоко нужно детализировать данные о затратах?
5. Что такое объект затрат?
6. Что такое центр затрат?
7. Что такое пользовательские категории UDP?
8. С какой целью создаются пользовательские категории UDP?
9. Какие пользовательские категории UDP могут быть использованы?
10. Как производится добавление категорий UDP?

Практическая работа №30.

Организационные диаграммы.

Цель: Научиться строить организационные диаграммы и диаграммы SwimLane в инструментальной среде Ramuseducational.

В результате освоения учебной дисциплины обучающийся должен **уметь:**

- ☐ использовать методы и критерии оценивания предметной области и методы определения стратегии развития бизнес-процессов организации;
- ☐ использовать и рассчитывать показатели и критерии оценивания информационной системы, осуществлять необходимые измерения.

В результате освоения учебной дисциплины обучающийся должен **знать:**

- ☐ требования к проектируемой системе;
- ☐ модели жизненного цикла информационной системы, методы проектирования информационной системы; технологии проектирования информационной системы, оценку и управление качеством информационной системы.

Теоретическая часть

Ramuseducational содержит набор инструментов для моделирования организационной структуры предприятия. **Ramuseducational** содержит четыре – словарь изображений (bitmap), словарь ресурсов, словарь ролей и словарь групп ролей.

Словарь изображений служит для импорта файлов в формате bmp в модель. Импортированные изображения можно использовать в диаграммах для улучшения их внешнего вида. Для импорта изображения следует перейти в меню *Dictionary/Bitmaps*. Появится диалог *Bitmap Dictionary* (рис.

1), в котором следует щелкнуть по кнопке *Import* и найти файл формата *bmp*.

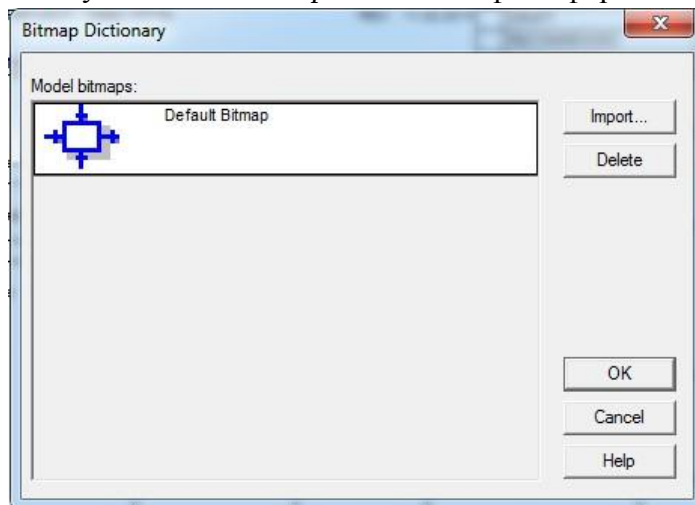


Рис.1 Диалоговое окно Bitmap Dictionary

Словарь Role Group Dictionary (меню *Dictionary/RoleGroup*) позволяет создать и определить свойства групп ролей. Группы ролей могут использоваться как на организационных диаграммах, так и на диаграммах Swim Lane. В качестве значения группы ролей может быть название предприятия, отдела, цеха или название региона, города и т. д.

Для каждой группы ролей может быть внесено описание, указано изображение, предварительно импортированное в словарь изображений, и указана важность группы ролей.

Словарь ролей может вызываться из меню *Dictionary/Role*.

Ролью может быть должность или позиция конкретного исполнителя. Каждой роли может соответствовать одна или несколько групп ролей. Кроме того, в словаре ролей для каждой роли можно внести определение (*Definition*), связать роль с изображением (*Bitmap*) и геометрической фигурой (*Shape*), указывать важность роли (*Importance*).

Словарь ресурсов (меню *Dictionary/Resource*) позволяет создать ресурс и связать его с комбинацией «группы ролей/роль». Ресурсом для роли может быть конкретный исполнитель. В качестве значения ресурса, например, можно использовать фамилию и имя сотрудника.

На основе информации, внесенной в словари изображений, группролей, ролей ресурсов, можно создать **организационную диаграмму**. Организационная диаграмма позволяет документировать и представить в виде дерева структуру организации (например штатное расписание и т.д.).

Для создания организационной диаграммы следует выбрать меню *Diagram/Add Organization Chart*. Появляется гид *Organization Chart Wizard*.

В первом диалоге гида следует внести название и имя автора диаграммы, группу ролей и роль для верхнего уровня иерархического дерева.

Второй диалог гида позволяет создать второй уровень иерархического дерева. Верхний список содержит все доступные роли с ассоциированными ресурсами, нижний – роли и ресурсы второго уровня иерархии. Кнопка *Add* позволяет перенести роли и ресурсы из верхнего списка в нижний, кнопка *Remove* – из нижнего в верхний.

Третий диалог *Organization Chart Wizard* предназначен для изменения свойств организационной диаграммы. В группе *Drawing* можно указать, какая именно информация будет отображаться на блоках диаграммы (наименование блока, имя группы ролей, роль и ресурс). Для отображения иконок на диаграмме в группе *Draw Style* следует выбрать опцию *Bitmap*. После щелчка по кнопке *Finish* создается организационная диаграмма. Для дополнения диаграммы следует щелкнуть правой кнопкой мыши по блоку и выбрать в контекстном меню один из пунктов:

- **Edit subordinate list** – редактирование блока;
- **Add subordinates** – добавляет нижний уровень;
- **Add sibling on left** – добавляет блок на текущий уровень слева от редактируемого блока;
- **Add sibling on right** – добавляет блок на текущий уровень справа от редактируемого блока.

Созданные в словаре *Role Dictionary* роли могут быть также использованы в диаграмме *Swim Lane*. **Диаграмма *Swim Lane* является разновидностью диаграммы *IDEF3***, позволяющей явно описывать роли и ответственности исполнителей в конкретной технологической операции. Эта диаграмма разделена на горизонтальные полосы, с каждой полосой может быть связана роль или *UDP* типа *Text List*. Полоса может содержать объекты диаграммы *IDEF3* (*UOW*, перекрестки и объекты ссылок), относящиеся к соответствующей роли. Для создания диаграммы *Swim Lane* следует выбрать меню *Diagram/Add SwimLanediagram*. Появляется гид *Swim Lane diagram Wizard*. В первом диалоге гида следует внести название и имя автора диаграммы, выбрать имя и номер

диаграммы IDEF3, на основе которой будет построена диаграмма, и группу ролей, из которой можно будет выбрать роли, связанные с диаграммой.

Во втором диалоге гида следует выбрать роли, на основе которых будет создана диаграмма. Диаграмма будет разделена на количество полос, указанных в колонке *Display Swim Lane*.

После щелчка по кнопке *Finish* создается новая диаграмма, все объекты которой расположены произвольно. Расположить объекты на полосах, соответствующих ролям, следует вручную.

Практическая работа №31. Диаграммы Swim Lane.

Цель: Научиться строить организационные диаграммы и диаграммы Swim Lane в инструментальной среде Ramus educational.

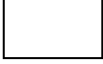
Практическая часть

Задание: Создать организационную диаграмму и диаграмму Swim Lane своей предметной области и на основе ранее созданных диаграмм декомпозиции, и диаграмм потоков работ.

Пример создания организационной диаграммы 1. Составим таблицу **Параметров полей словаря групп ролей.**

Наименование (Name)	Определение (Definition)	Важность (Importance)	Фигура (Share)
---------------------	--------------------------	-----------------------	----------------

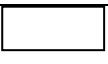
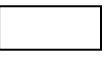
Name	Definition	Bitmap	Importance	Shape
Kommercheskaia slugba	Sklad gotovoi produkcii		Medium	□
Osnovnoe proizvodstvo	Sborochnoe		Medium	□
Poddergivaussee proizvodstvo	Transportni cex		Medium	◇
Slugba upravleniia personalom	Otдел kadrov		Medium	□
Upravlenie	Centralnaia buhgalleria		High	◇
			Low	□

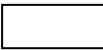
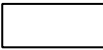
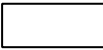
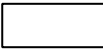
Коммерческая служба	Склад готовой продукции	Medium	
Основное производство	Сборочное	Medium	
Поддерживающее производство	Транспортный цех	Medium	
Служба управления персоналом	Отдел кадров	Medium	
Управление	Главная бухгалтерия	High	

2. Вызовите словарь групп ролей **Group Dictionary** (меню **Dictionary/RoleGroup**), в результате откроется словарь групп ролей.

3. Заполните словарь соответственно таблице.

4. Составим таблицу **Параметров словаря ролей**.

Наименование (Name)	Группа (Role Group)	Важность (Importance)	Фигура (Share)
Генеральный директор	Управление	High	
Главный бухгалтер	Поддерживающее производство	Medium	
Директор по качеству	Управление	High	
Директор по производству	Управление	High	
Заведующий бюро материально-технического снабжения	Коммерческая служба	Medium	
Заведующий складом	Коммерческая служба	Low	
Заместитель генерального директора	Управление	High	
Инспектор контроля качества	Поддерживающее производство	Medium	

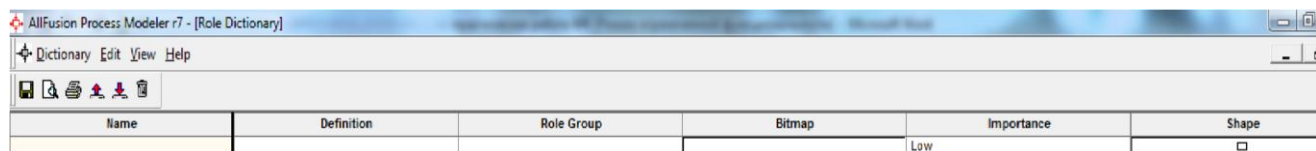
Кадровик	Служба управления персоналом	Low	
Коммерческий директор	Управление	High	
Мастер сборочного цеха	Основное производство	Low	
Мастер упаковочного цеха	Основное производство	Low	
Настройщик технологического процесса	Основное производство	Low	
Оператор технического оборудования	Основное производство	Low	
Оператор сборочного оборудования	Основное производство	Low	
Менеджер по продажам	Коммерческая служба	Medium	
Начальник Службы управления персоналом	Служба управления персоналом	Medium	
Начальник Юридического отдела	Поддерживаю щее производство	High	
Финансист	Поддерживаю щее производство	Medium	
Финансовый директор	Управление	High	
Экономист	Поддерживаю щее производство	Medium	

Практическая работа №32.

Заполнение словаря ролей.

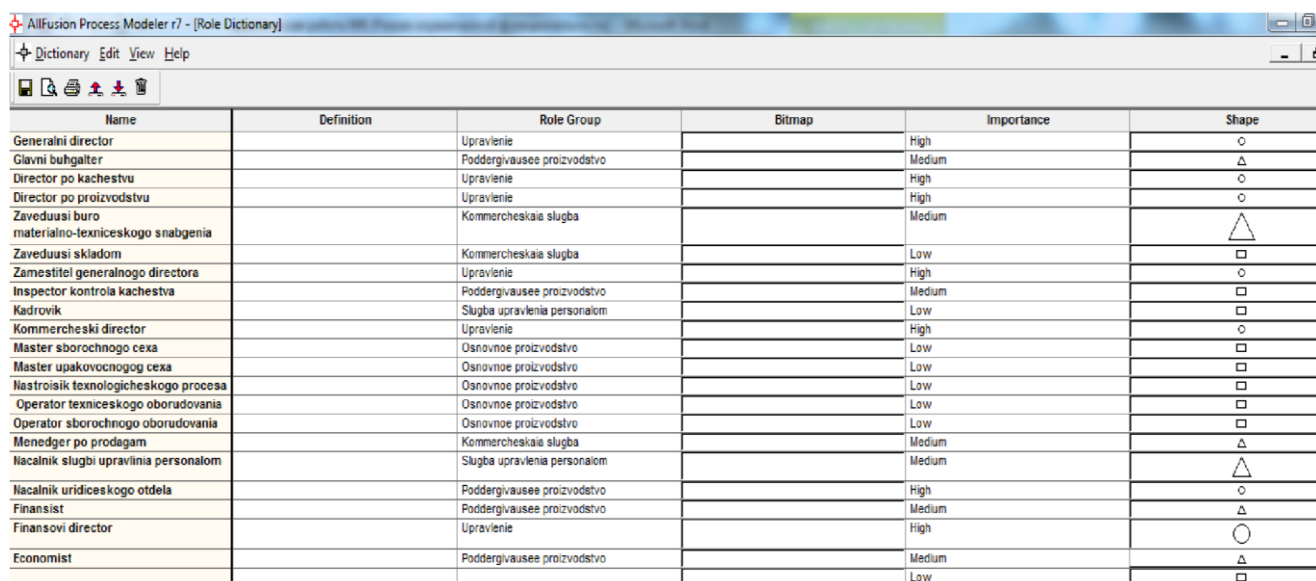
Цель: Научиться строить организационные диаграммы и диаграммы SwimLane в инструментальной среде Ramuseducational.

5. Вызовите словарь ролей **RoleDictionary** (меню **Dictionary/Role**) в результате откроется словарь ролей.



Name	Definition	Role Group	Bitmap	Importance	Shape
				Low	□

6. Заполните словарь ролей согласно заполненной таблице.



Name	Definition	Role Group	Bitmap	Importance	Shape
Generalni direktor		Upravlenie		High	○
Glavni buhgalter		Poddergiavusee proizvodstvo		Medium	△
Director po kachestvu		Upravlenie		High	○
Director po proizvodstvu		Upravlenie		High	○
Zaveduusi buro materialno-tekhnicheskogo snabzhenia		Kommercheskaia slugba		Medium	△
Zaveduusi skladom		Kommercheskaia slugba		Low	□
Zamestitel generalnogo direktora		Upravlenie		High	○
Inspector kontrolya kachestva		Poddergiavusee proizvodstvo		Medium	□
Kadrovik		Slugba upravleniya personalom		Low	□
Kommercheskiy direktor		Upravlenie		High	○
Master sborochnogo cexa		Osnovnoe proizvodstvo		Low	□
Master upakovocnogo cexa		Osnovnoe proizvodstvo		Low	□
Nastroisik tekhnologicheskogo protsessa		Osnovnoe proizvodstvo		Low	□
Operator tekhnicheskogo oborudovaniya		Osnovnoe proizvodstvo		Low	□
Operator sborochnogo oborudovaniya		Osnovnoe proizvodstvo		Low	□
Menedzher po prodazham		Kommercheskaia slugba		Medium	△
Nachalnik sluzhby upravleniya personalom		Slugba upravleniya personalom		Medium	△
Nachalnik uridicheskogo otdela		Poddergiavusee proizvodstvo		High	○
Finansist		Poddergiavusee proizvodstvo		Medium	△
Finansoviy direktor		Upravlenie		High	○
Economist		Poddergiavusee proizvodstvo		Medium	△
				Low	□

7. Составим таблицу **Параметров словаря ресурсов**

Наименование (Name)	Ассоциации (Associations)
Акимова В. Д.	Управление / Генеральный директор
Апаксин С. М.	Управление / Финансовый директор
Байгулова О. И.	Служба управления персоналом / Начальник службы управления персоналом
Буравин М. Б.	Управление / Коммерческий директор
Владина М. П.	Управление / Директор по качеству

Гончар Р. С.	Коммерческая служба / Менеджер по продажам
Горин М. П.	Управление / Директор по производству
Зайцева П. В.	Основное производство / Мастер сборочного цеха
Кирилина Т. Е.	Управление / Заместитель генерального директора
Козлов А. А.	Поддерживающее производство / Начальник Юридического отдела
Крымова П. Р.	Поддерживающее производство / Финансист
Лант Ю. Т.	Основное производство / Мастер упаковочного цеха
Лапин Т. Т.	Поддерживающее производство / Экономист
Рисинь С. В.	Поддерживающее производство / Главный бухгалтер
Чупрова Е. Ю.	Коммерческая служба / Заведующий бюро материально- технического снабжения
Шапор Д. С.	Поддерживающее производство / Инспектор контроля качества
Щипова С. Б.	Коммерческая служба / Заведующий складом
Ключников М. А	Основное производство / Настройщик технологических процессов
Крылова М.Н.	Основное производство / Оператор технического оборудования
Афанасьева Э. М.	Основное производство / Оператор сборочного оборудования

8. Вызовите словарь ресурсов **Resource Dictionary** (меню **Dictionary/Resource**).



9. Заполните словарь ресурсов согласно заполненной таблице

AllFusion Process Modeler v7 - [Resource Dictionary]		
Dictionary Edit View Help		
Name	Definition	Associations
Akimova V.D.		Upravlenie / Generalni director
Apraksin S.M.		Upravlenie / Finansovii director
Baigulova O.I.		Sluzhba upravleniya personalom / Nacelnik sluzhbi upravleniya personalom
Baravin M.B.		Upravlenie / Kommercheskii director
Vladina M.P.		Upravlenie / Director po kachestvu
Goncar R.S.		Kommercheskaya sluzhba / Menedzher po prodazham
Gorin M.P.		Upravlenie / Director po proizvodstvu
Zaiceva P.V.		Osnovnoe proizvodstvo / Master sbornogo cexa
Kirilina T.E.		Upravlenie / Zamestitel generalnogo direktora
Kozlova A.A.		Podderglivayushcheye proizvodstvo / Nacelnik urudiceskogo otdela
Krimova P.R.		Podderglivayushcheye proizvodstvo / Finansist
Lant U.T.		Osnovnoe proizvodstvo / Master upakovonogo cexa
Lapin T.T.		Podderglivayushcheye proizvodstvo / Ekonomist
Risin S.V.		Podderglivayushcheye proizvodstvo / Glavni buhgalter
Cuprova E.I.		Kommercheskaya sluzhba / Zaveduushi buro materialno-tekhnicheskogo snabzheniya
Sapor D.S.		Podderglivayushcheye proizvodstvo / Inspector kontrolya kachestva
Sipova S.B.		Kommercheskaya sluzhba / Zaveduushi skladom
Lisov A.S.		Osnovnoe proizvodstvo / Master upakovonogo cexa
Klucnikov M.A.		Osnovnoe proizvodstvo / Nastroisk tekhnologicheskogo procesa
Krilova M.I.		Osnovnoe proizvodstvo / Operator tekhnicheskogo oborudovaniya
Afanasieva E.M.		Osnovnoe proizvodstvo / Operator sbornogo oborudovaniya

10. Выберите меню **Diagram/Add Organization Chart**. Появится **Organization Chart Wizard**. В первом диалоге введите название и имя автора диаграммы, группу ролей и роль для верхнего уровня иерархического дерева.

Organization Chart Wizard - Step 1 of 3

- You must name your diagram.
Name: Структурная схема ОАО "Завод металлоконструкций"
- Choose a Role Group you have already defined in the Role Group Dictionary. A Role Group represents the grouping criterion of Roles in the organization.
Role Group: Upravlenie
- Choose a Role you have already defined in the Role Dictionary to be the top level Role in the Organization Chart.
Role: Generalni director
- Choose the resource you have already defined in the Resource Dictionary that you associate with the selected Role from above.
Resource: Akimova V.D.
- (optional) Enter the diagram author name.
Author: Назарова О.И.

* Note: If you do not choose a Role Group, you will get an empty box.

< Назад Далее > Готово Отмена Справка

Практическая работа №3.3

Создание второго уровня диаграммы.

Цель: Научиться строить организационные диаграммы и диаграммы SwimLane в инструментальной среде Ramus educational.

11. Создайте второй уровень диаграммы

Organization Chart Wizard - Step 2 of 3

Role Group for the next level:
Upravlenie

Available Role / Resources:

- < Empty Box >
- Director po kachestvu
- Director po kachestvu -- Vladina M.P.
- Director po proizvodstvu
- Director po proizvodstvu -- Gorin M.P.
- Finansovii director
- Finansovii director -- Apraksin S.M.
- Generalni director
- Generalni director -- Akimova V.D.

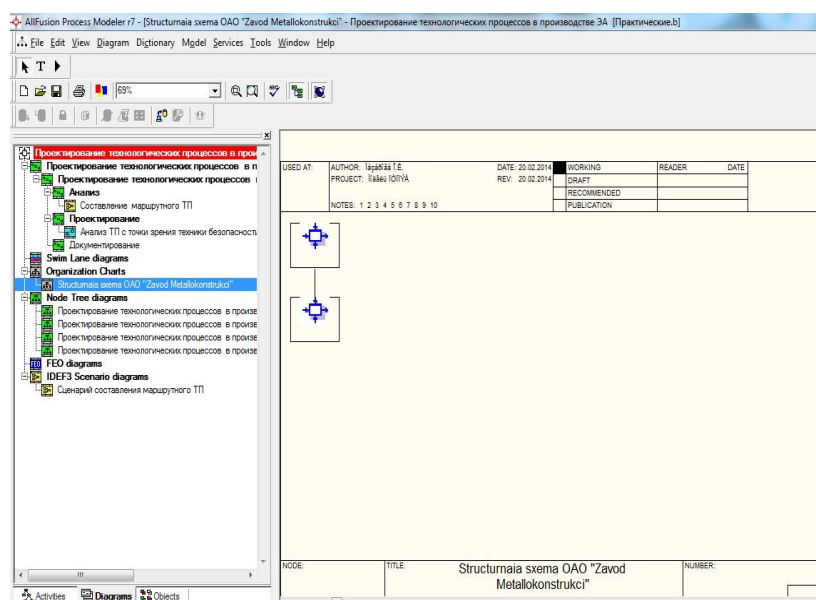
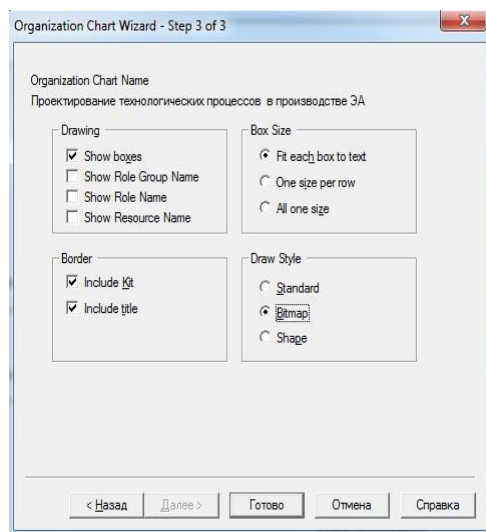
Add: ↓ Remove: ↑

Selected Group/Role/Resources:
Upravlenie -- Zamestitel generalnogo direktora -- Kirilina T.E.

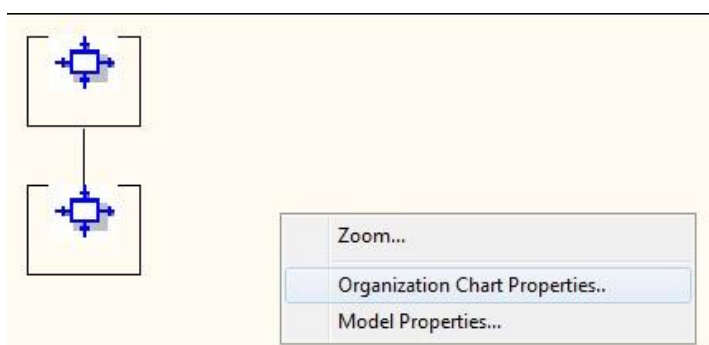
Order ↑ ↓

< Назад Далее > Готово Отмена Справка

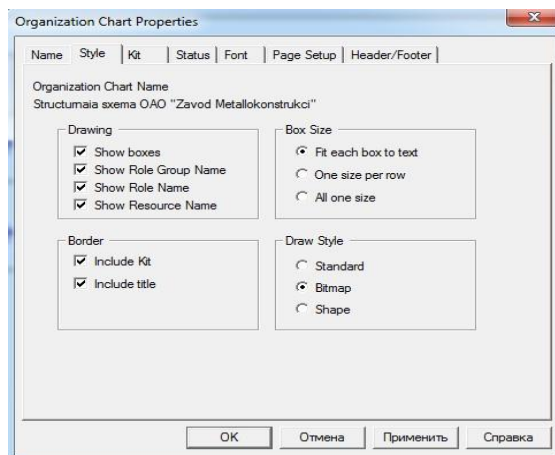
12. Третий диалог **OrganizationChartWizard** предназначен для изменения свойств организационной диаграммы. В группе **Drawing** можно указать, какая именно информация будет отображаться на блоках диаграммы (наименование блока, имя группы ролей, роль и ресурс). Для отображения иконок на диаграмме в группе **DrawStyle** следует выбрать опцию **Bitmap**. После щелчка по кнопке **Finish** создаётся организационная диаграмма.



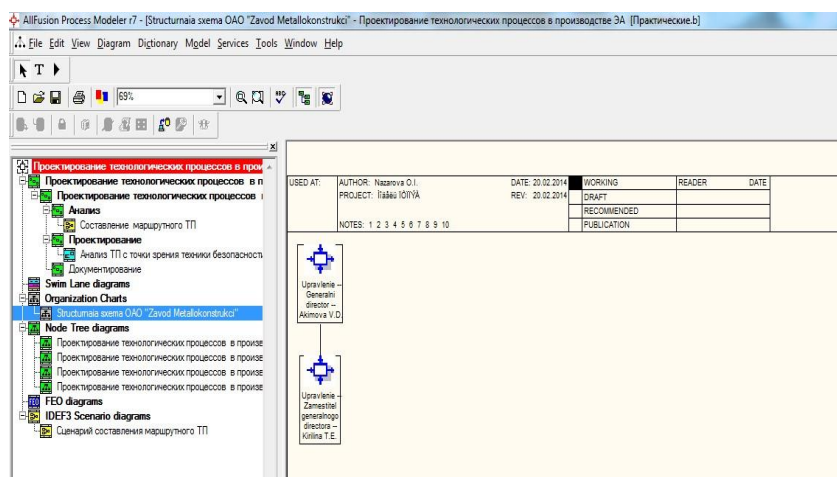
13. Щелкните правой клавишей мыши по рабочей области и в появившемся меню выберите **Organization Chart Properties**



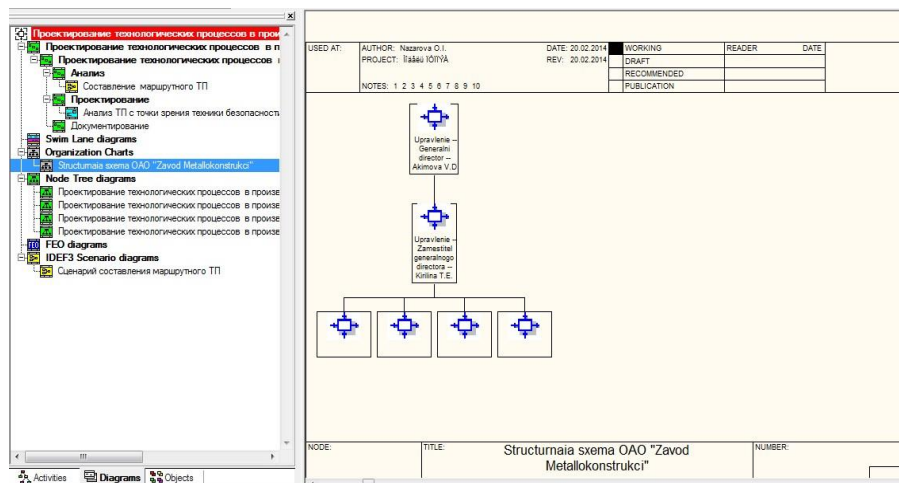
14. В открывшемся диалоговом окне перейдите на вкладку **Style** и выделите все пункты в области настройки изображения (**Drawing**)



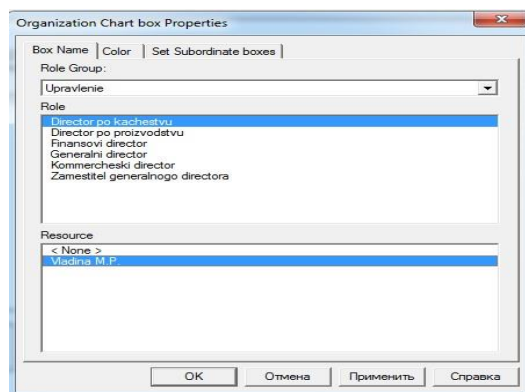
15. Нажмите кнопку **Применить** и закройте окно настройки свойств, в результате диаграмма примет вид, показанный на рисунке.



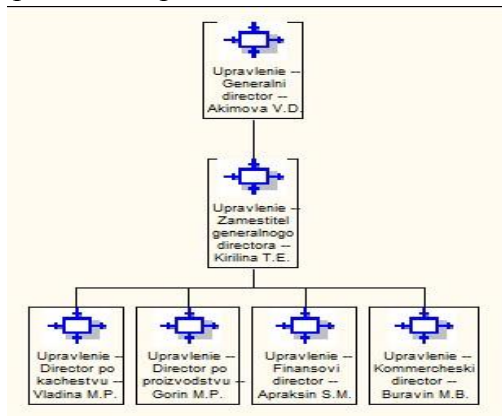
16. Щелкните Правой клавишей мыши по блоку **Управление – Заместитель генерального директора – Кирилина Т. Е.**, в появившемся меню выберите **Add subordinates**. В появившемся диалоговом окне укажите количество добавляемых блоков «4» и нажмите **ОК**. На третий уровень диаграммы будет добавлено еще 4 блока



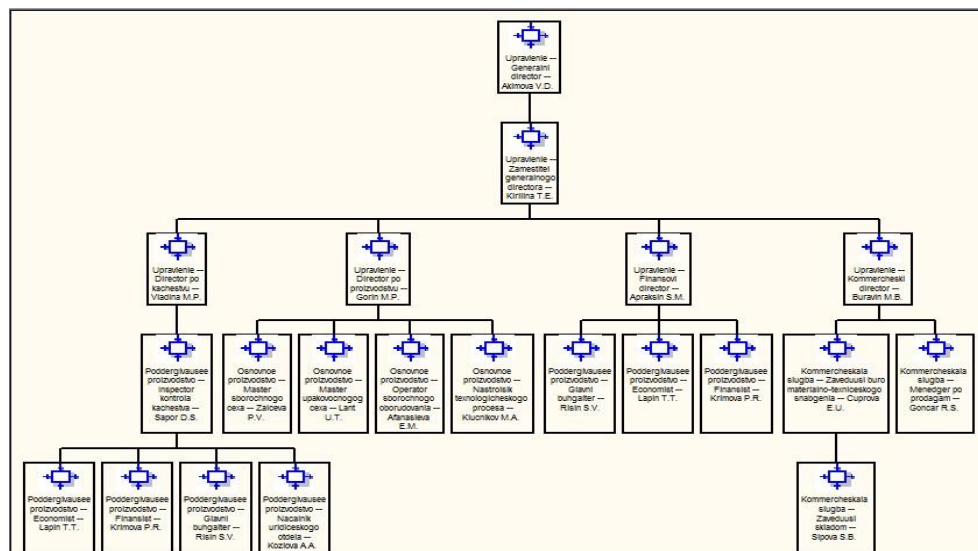
17. Щелкните правой клавишей мыши по левому нижнему блоку диаграммы и выберите пункт меню «**Role Name...**», откроется диалоговое окно. В этом окне укажите группу ролей, роль, ресурс.



18. Аналогичным образом настраиваются и остальные блоки



19. Далее подобным образом добавляются остальные уровни диаграммы. Законченная диаграмма представлена на рисунке.

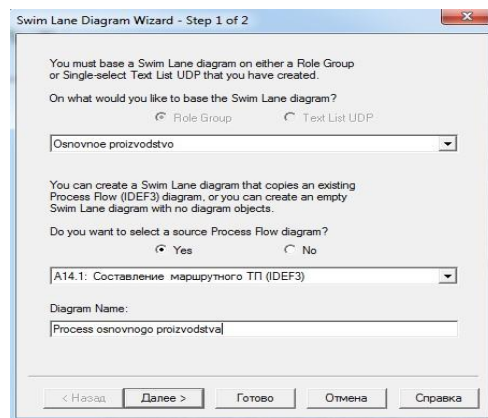


Пример создания диаграммы Swim Lane

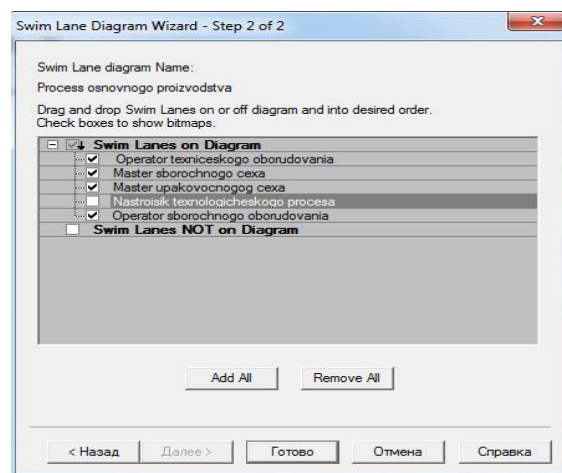
Цель: Научиться строить организационные диаграммы и диаграммы SwimLane в инструментальной среде Ramuseducational.

Пример создания диаграммы Swim Lane

1. Для создания диаграммы **Swim Lane** выберите меню **Diagram/Add Swim Lane diagram**. Появляется гид **Swim Lane diagram Wizard**.
2. В первом диалоге гида следует внести название и имя автора диаграммы, выбрать имя и номер диаграммы IDEF3, на основе которой будет построена диаграмма, и группу ролей, из которой можно будет выбрать роли, связанные с диаграммой.



3. Во втором диалоге гида следует выбрать роли, на основе которых будет создана диаграмма. Диаграмма будет разделена на количество полос, указанных в колонке **Display Swim Lane**.

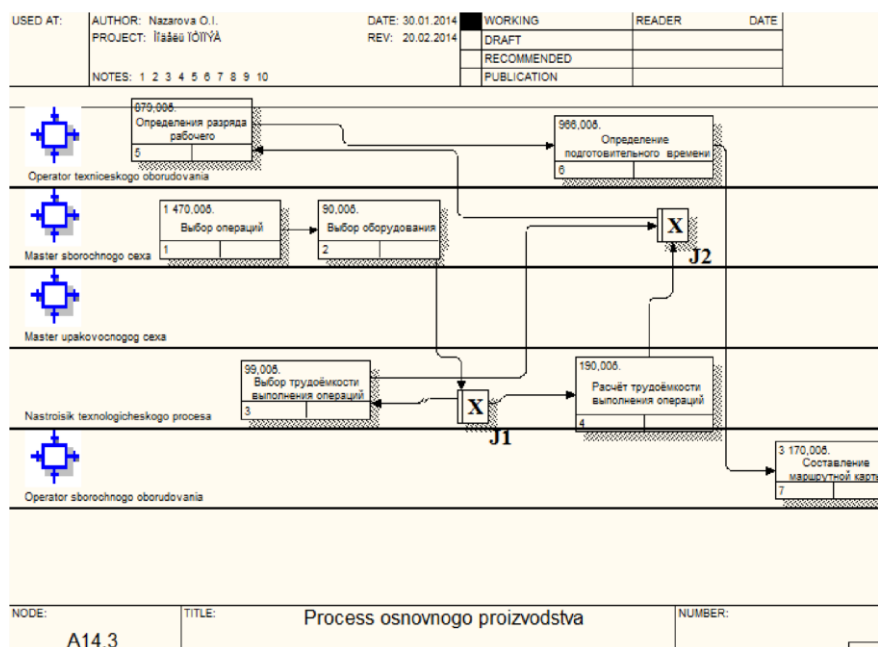


4. После щелчка по кнопке **Готово** создается новая диаграмма, все объекты которой расположены произвольно. Расположить объекты на полосах, соответствующих ролям, следует вручную.

Практическая работа №35.

Характеристики диаграмм SwimLane.

Цель: Научиться строить организационные диаграммы и диаграммы SwimLane в



инструментальной среде Ramuseducational.

Контрольные вопросы:

1. Для чего используются организационные диаграммы?
2. Каков порядок построения организационной диаграммы?
3. Насколько глубоко можно детализировать организационную диаграмму?
4. Какой глубины детализация необходима при построении организационной диаграммы?
5. С какого уровня иерархии сотрудников допустимо начинать организационную диаграмму?
6. Для чего используются диаграммы Swim Lane?
7. Как строятся диаграммы Swim Lane?
8. На основе какой диаграммы строится диаграмма Swim Lane?
9. Чем отличаются IDEF3 и Swim Lane диаграммы?
10. Где должны располагаться перекрестки на диаграмме Swim Lane?

Основная литература:

1. Яхьяева, Г.Э. Основы теории нейронных сетей / Г.Э. Яхьяева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 200 с. : ил. - (Основы информационных технологий). - ISBN 978-5-94774-818-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429110>
2. Лягинова, О.Ю. Разработка схем и диаграмм в Microsoft Visio 2010 / О.Ю. Лягинова. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 128 с. : схем., ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428810>
3. Кияев, В.И. Развитие информационных технологий / В.И. Кияев, О.Н. Граничин. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 199 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL:

литература:

1. Фуфаев Э.В. Разработка и эксплуатация удаленных баз данных: учебник для студ. учреждений сред.проф. образования/ Э.В.Фуфаев, Д.Э. Фуфаев. – 4-е изд., стер. – М.: Издательский центр «Академия», 2014. – 256 с.
2. Волкова, Т.В. Основы проектирования компонентов автоматизированных систем: учебное пособие / Т.В. Волкова ; Министерство образования и науки Российской Федерации, Оренбургский Государственный Университет, Кафедра программного обеспечения вычислительной техники и автоматизированных систем. - Оренбург : ОГУ, 2016. - 226 с. : табл., ил. - Библиогр. в кн. - ISBN 978-5-7410-1560-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=471129>
3. Леоненков, А. Визуальное моделирование в среде IBM Rational Rose 2003 / А. Леоненков. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 193 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429149>