

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ В УПРАВЛЕНИИ И
ПРОЕКТИРОВАНИЯ (КСУП)

А. А. Изюмов

Автоматизация бизнес-процессов

Методические указания к лабораторным и самостоятельным работам

Томск
2023

Изюмов А. А.

Автоматизация бизнес-процесов : методические указания к лабораторным и самостоятельным работам / А. А. Изюмов – Томск : КСУП, ТУСУР, 2023. – 64 с.

Данное учебное пособие предназначено для бакалавров по направлениям 27.03.04 «Управление в технических системах», 27.04.04 «Управление в технических системах», 09.03.01 «Информатика и вычислительная техника», 09.04.01 «Информатика и вычислительная техника». Также пособие полезно для преподавателей и студентов высших учебных заведений технических направлений подготовки.

В пособии представлены методы построения UML-диаграмм, основы работы в 1С:Предприятии и построения на её базе конфигурации.

Содержание

| | |
|---|-----------|
| Содержание | 3 |
| Введение | 5 |
| 1 Организация самостоятельной работы | 6 |
| 1.1 Рабочая программа и план обучения..... | 6 |
| 1.2 Теоретический материал..... | 7 |
| 1.3 Рекомендации по работе с учебной и научной литературой | 8 |
| 1.4 Консультации..... | 10 |
| 1.5 Практические и лабораторные работы..... | 11 |
| 1.6 Контрольные мероприятия..... | 12 |
| 1.7 Итоговая аттестация..... | 13 |
| Лабораторная работа №1 «Поиск и формализация данных» | 15 |
| 2.1 Понятие бизнес-процесса | 15 |
| 2.2 Задание | 28 |
| 2.3 Порядок выполнения работы | 28 |
| 2.4 Содержание отчета..... | 29 |
| Лабораторная работа №2 «Основы работы в 1С. Справочники»..... | 29 |
| 3.1 Основы работы в 1С | 29 |
| 3.2 Задание | 37 |
| 3.3 Порядок выполнения работы | 37 |
| 3.4 Содержание отчета..... | 45 |
| Лабораторная работа №3 «Основы работы в 1С. Документы. Проведение» | 46 |
| 4.1 Основы работы с документами в 1С | 46 |
| 4.2 Задание | 51 |
| 4.2.1 Задание | 51 |
| 4.3 Порядок выполнения работы | 51 |
| 4.4 Содержание отчета..... | 52 |
| Лабораторная работа №4 «Основы работы в 1С. Документы. Проведение» | 53 |
| 5.1 Основы работы с печатными формами..... | 53 |
| 5.2 Задание | 56 |

| | |
|---|-----------|
| 5.2.1 Задание №1 | 56 |
| 5.2.2 Порядок выполнения работы | 56 |
| 5.2.2 Задание №2..... | 60 |
| 5.2.3 Порядок выполнения работы | 60 |
| 5.4 Содержание отчета..... | 61 |
| Список литературы..... | 63 |
| Дополнительные источники для организации самостоятельного обучения | 64 |

Введение

Предлагаемый курс лабораторных работ предназначен для студентов по направлениям подготовки 27.03.04 «Управление в технических системах», 27.04.04 «Управление в технических системах», 09.03.01 «Информатика и вычислительная техника», 09.04.01 «Информатика и вычислительная техника», 15.03.04 «Автоматизация технологических процессов и производств» позволяет сформировать у студентов необходимые компетенции для самостоятельного углубленного изучения предметной области. Целью предлагаемых методических указаний является знакомство с порядком организации самостоятельной работы по изучению дисциплины «Автоматизация бизнес-процессов».

Самостоятельная работа – основной вид учебной деятельности студентов ТУСУР, который позволяет овладевать фундаментальными знаниями, профессиональными умениями и навыками в конкретных областях науки, получать опыт творческой и исследовательской деятельности; развивать такие качества личности, как самостоятельность в принятии решений, ответственность, организованность; использовать творческий подход к решению проблем учебного и профессионального уровней.

Задачи самостоятельной работы студента:

- систематизация и закрепление полученных теоретических знаний и практических умений;
- углубление и расширение теоретической подготовки; формирование умения использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;

- использование материала, собранного и полученного в ходе самостоятельных занятий, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к зачетам и экзаменам.

Основные черты самостоятельной деятельности в высшей школе:

- *целенаправленность*, т. е. такая активность, которая направлена на достижение сознательно поставленной цели;
- *продуманность*, т. е. анализ ситуации и выбор способов и средств достижения цели и последовательности будущих действий;
- *осознанность*, планирование и предвидение возможного результата, наличие логической схемы;
- *структурность*, специфический набор действий и последовательность их осуществления;
- *результативность*, когда деятельность находит свое завершение в результате.

1 Организация самостоятельной работы

1.1 Рабочая программа и план обучения

Рабочая программа дисциплины – программа освоения учебного материала, соответствующая требованиям государственного образовательного стандарта высшего образования и учитывающая специфику подготовки студентов по выбранному направлению.

В рабочей программе дана общая информация по дисциплине: цели и задачи дисциплины, место в образовательной программе, требование к результатам освоения дисциплины (на формирование каких компетенций направлена), объем и виды учебной работы, формы аттестации, основная и дополнительная литература, примерные оценочные материалы.

Рабочая программа позволяет представить объем предстоящей работы по дисциплине и составить план самостоятельного обучения. План должен включать следующие важные составляющие:

- 1) изучение теоретического материала дисциплины, освоение терминологии, ознакомление с классификациями и др. (для этого необходимо оценить объем материалов электронного курса и дополнительной литературы);
- 2) подготовка и выполнение контрольных работ;
- 3) подготовка и прохождение итоговой аттестации по дисциплине.

Рабочая программа дисциплины доступна из рабочего учебного плана соответствующей основной профессиональной образовательной программы (<https://edu.tusur.ru/opops>).

1.2 Теоретический материал

Учебное пособие (теоретический материал электронного курса) содержит основную теоретическую информацию по дисциплине. Деление на главы (темы) позволяет проследить логику изложения материала и равномерно распределить его в своем плане обучения. Составление конспектов по каждой теме поможет закрепить материал. Особое внимание следует обращать на блоки теоретического материала, выделенные пиктограммами. Данный материал обязателен для усвоения и запоминания. Рекомендуется уделить внимание содержанию таблиц и рисункам, являющимся источниками важной наглядной и структурированной информации.

Самостоятельная работа предполагает обязательное изучение рекомендованной литературы, выполнение практических заданий и закрепление усвоенного материала. Для углубленного изучения тем рекомендуется использовать дополнительные литературные источники.

Студенты ТУСУР имеют доступ к полным текстам изданий электронных библиотечных систем «[ЮРАЙТ](#)» и «[ЛАНЬ](#)».

1.3 Рекомендации по работе с учебной и научной литературой

Самостоятельная работа по освоению теоретического материала учебного пособия и другой учебной и научной литературы (а также самостоятельное теоретическое исследование изучаемых проблем) – это важнейшее условие формирования научного способа познания. Таким образом, чтение научного текста является частью познавательной деятельности, главная цель которой – извлечение из текста необходимой информации. Осознание читающим внутренней установки при обращении к печатному слову (выделение нужных сведений, усвоение информации полностью или частично, критический анализ материала и т. п.) определяет эффективность осуществляемого действия.

Грамотная работа с книгой, особенно если речь идет о научной литературе, предполагает *соблюдение ряда правил*, для овладения которыми необходимо учиться. Прежде всего, при такой работе невозможен формальный, поверхностный подход. Главное правило при работе над книгой – не механическое заучивание, не простое накопление цитат, выдержек, а сознательное усвоение прочитанного, осмысление его, стремление дойти до сути.

Изучение книги должно происходить в определенной последовательности. Вначале следует ознакомиться с оглавлением (содержанием), предисловием, введением. Это позволяет получить общее представление о структуре и вопросах, которые рассматриваются в книге. Затем можно переходить непосредственно к чтению. Чтобы получить цельное представление о книге, первый раз стоит прочитать ее от начала до конца. При повторном чтении происходит постепенное глубокое осмысление каждой главы, критического материала и позитивного изложения; выделение основных идей, системы аргументов, наиболее ярких примеров и т. д. Непременным правилом чтения должно быть выяснение незнакомых слов, терминов, выражений, неизвестных имен, названий. Важная роль в связи с этим принадлежит библиографической подготовке студентов. Она включает в себя умение активно, быстро пользоваться научным аппаратом книги, справочными изданиями, каталогами; вести поиск необходимой информации, обрабатывать и систематизировать ее.

Рекомендуется использовать четыре основные установки в чтении научного текста:

- информационно-поисковая (задача – найти, выделить искомую информацию);
- усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения, излагаемые автором, так и всю логику его рассуждений);
- аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему);
- творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т. п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Научная методика работы с литературой предусматривает фиксирование прочитанной информации в систематизированных записях разного рода. Это позволяет привести в систему знания, полученные при чтении, сосредоточить внимание на главных положениях, зафиксировать, закрепить их в памяти, а при необходимости вновь обратиться к ним.

Основные виды систематизированной записи прочитанного:

- тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
- цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;
- конспектирование – краткое и последовательное изложение содержания прочитанного.

Методические рекомендации по составлению конспекта

Внимательно прочитайте текст. Уточните в глоссарии или справочной литературе непонятные слова. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов,

последовательно раскрываемых затем в конспекте. Это первый элемент конспекта.

Вторым элементом конспекта являются тезисы. Тезис – это кратко сформулированное положение. Для лучшего усвоения и запоминания материала следует записывать тезисы своими словами. Тезисы, выдвигаемые в конспекте, нужно доказывать.

Третий элемент конспекта – основные доводы, доказывающие истинность рассматриваемого тезиса. В конспекте могут быть положения и примеры.

Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного.

Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнений и дополнений необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы. Конспект ускоряет повторение материала, экономит время при повторном обращении к работе.

1.4 Консультации

В связи с тем, что не всегда возможно самостоятельно найти ответы на возникшие в ходе изучения дисциплины вопросы, на факультете существует возможность задать вопрос преподавателю, записавшись к нему на консультацию. Вопросы можно задавать как по теоретическому материалу, так и по подготовке, оформлению отчетов любого вида работ.

Предусмотрено несколько ресурсов, где можно задать вопрос:

- на учебном форуме из страницы электронного курса;

– написав письмо с любого другого почтового сервера на адрес преподавателя.

Также существует возможность обсудить любой учебный вопрос на форуме с другими студентами (внутри электронного курса), коллективно найти решение. Однако следует помнить, что мнения других студентов могут быть ошибочными, и критически относиться к высказанным идеям.

1.5 Практические и лабораторные работы

Лабораторные и практические работы являются основными видами деятельности, направленными на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

Выполнение студентами лабораторных и практических работ необходимо:

- для формирования практических умений в соответствии с требованиями к уровню подготовки студентов, установленными рабочей программой дисциплины по конкретным разделам (темам);
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- совершенствования умений применять полученные знания на практике, реализации единства интеллектуальной и практической деятельности;
- развития интеллектуальных умений;
- выработки при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Практическая работа

В курсе может предусмотрено выполнение самостоятельной практической работы. Ее целью является развитие умений и закрепление навыков в решении учебных и профессиональных задач, подготовке к промежуточной и итоговой

аттестации. Перед тем как приступить к выполнению работы, следует ознакомиться с образцом решения, предоставленным преподавателем.

Лабораторная работа

Задание к текстовой лабораторной работе располагается в учебных материалах. Для выполнения работы необходимо повторить уже усвоенные разделы учебного пособия по теме, изучить дополнительный материал, который приведен в методических указаниях и дополнительной литературе. Для успешного выполнения лабораторных работ рекомендуется:

- подробно изучить текст задания и определить свой вариант;
- обязательно оформить отчет в соответствии с представленными требованиями;
- отправить его на проверку преподавателю.

В отличие от самостоятельной практической работы лабораторные работы оцениваются, получение зачета является обязательным условием для аттестации по дисциплине.

1.6 Контрольные мероприятия

Самоконтроль

Самоконтроль – один из важнейших факторов, обеспечивающих самостоятельную деятельность обучающихся. Самоконтроль необходим для проверки остаточных знаний по итогам изучения каждой темы. Большинство курсов снабжены двумя уровнями самоконтроля. В текстовом материале после каждой главы представлен перечень вопросов и заданий, позволяющих проконтролировать усвоение ключевых вопросов темы. Если ответы на вопросы вызывают затруднение, следует еще раз перечитать материал.

Второй уровень самоконтроля – тестовые задания в электронном курсе. Тестирование является наиболее эффективной формой контроля и имеет ряд преимуществ перед другими формами контроля:

- затрата небольшого количества времени на выполнение тестовых заданий;
- возможность оперативного получения информации о степени усвоения знаний;
- возможность оперативной корректировки знаний и умений и др.

Формат электронных тестов максимально приближен к тем заданиям, которые необходимо будет выполнить на итоговой аттестации по дисциплине.

Контрольная работа

Цель выполнения контрольной работы – закрепить знания, полученные студентами при изучении теоретического материала, а также, в ряде случаев, отработать навыки решения практических задач. К выполнению контрольной работы следует серьезно подготовиться, повторив весь теоретический материал и потренировавшись в решении задач.

Учебный план дисциплины «Автоматизация бизнес-процессов» предполагает выполнение двух компьютерных тестовых контрольных работ, задания на которые размещены в электронном курсе.

В отличие от самоконтроля, контрольные работы оцениваются, получение зачета является обязательным условием для аттестации по дисциплине.

1.7 Итоговая аттестация

Для получения зачета достаточно успешно выполнить все оцениваемые работы по дисциплине «Автоматизация бизнес-процессов».

Наиболее ответственным этапом в обучении студентов является экзаменационная сессия, во время нее студенты отчитываются о выполнении учебной программы, об уровне и объеме полученных знаний. Это

государственная отчетность студентов за изучение учебной дисциплины, поэтому ответственность за успешную сдачу экзаменационной сессии велика. На сессии студенты сдают экзамены и дифференцированные зачеты.

Лабораторная работа №1 «Поиск и формализация данных»

Цель работы: создание BPMN-диаграммы.

Темы для предварительного изучения: Понятие бизнес-процесса. UML. BPMN.

2.1 Понятие бизнес-процесса

В теории автоматизация бизнес-процессов и автоматизация бизнеса — равнозначные понятия, потому что в бизнесе всё есть процесс, который можно разделить на подпроцессы и этапы и описать. Но в реальном мире в компаниях есть ряд активностей с высоким уровнем социальности, которые могут быть автоматизированы лишь частично. Например, корпоративная культура, подбор и управление персоналом, управление компанией в целом — эти сферы бизнеса предполагают вовлечение человека не только как ресурса, но и как личности. Таким образом, бизнес-активности, которые проблематично разложить на процессы, но можно частично автоматизировать попадают в зону автоматизации бизнеса, что более широким понятием, чем автоматизация бизнес-процессов.

Система автоматизации бизнеса — это всё то, что превращает часть ручного труда в машинный, автоматический, переводит в сферу инженерных и информационных технологий. По большому счету, к технологиям автоматизации бизнеса можно отнести огромный пласт средств, от производственного оборудования и робототехники, до ПО и каналов связи.

Термин «бизнес-процесс» используется для описания организации и структуры взаимосвязанных действий сотрудников компании для успешного достижения поставленной цели. Суть автоматизации бизнес-процесса состоит в том, чтобы задать структуру передачи ответственности за выполнение процесса и, следовательно, достижения цели. Для отдельных сотрудников бизнес-процесс представляет собой шаги для достижения успешных результатов. В случае малых и крупных групп бизнес-процесс понимается как назначение

обязанностей, которые каждый участник группы должен выполнить для выполнения общей для группы задачи.

Залогом успеха бизнес-процесса являются определенные незаменимые компоненты. Прежде всего, для каждого этапа процесса требуются данные и материалы. Во-вторых, необходим алгоритм – процесс, которому люди или программное обеспечение (в случае автоматизации бизнес-процесса) могли бы следовать. Наконец, должен быть определён результат, с целью достижения которого прилагаются усилия, тратится время и используются данные. Необходимо признать важность этих компонентов автоматизации бизнес-процесса и уделить время для выбора и формализации каждого из них, поскольку, только заложив правильную основу, вы сможете рассчитывать на корректный результат.

Ключевая идея использования цифровых инструментов состоит в том, чтобы эффективно работать со всеми данными о клиентах, задачах и услугах/товарах компании, а также автоматизировать определенные действия, которые могут быть выполнены без человеческих усилий. Целью автоматизации бизнес-процессов не является замена сотрудников, которые выполняют повторяющиеся задачи, на программных роботов. Благодаря автоматизации бизнес-процессов, сотрудники компании избавляются от рутинных и механических задач и получают возможность сфокусироваться на выполнении новых обязанностей, исследовательских задач или любой активности, предполагающей применение опыта и знаний сотрудника. Автоматизация бизнес-процессов также помогает группам людей лучше координировать усилия, поэтому дезорганизация может быть исключена из уравнения. Наконец, автоматизация бизнес-процессов может гарантировать более высокую точность результатов. Все эти преимущества являются причиной того, что все больше и больше предпринимателей, руководителей отделов, а также малых и крупных компаний применяют автоматизацию бизнес-процессов в своей работе.

Поскольку бизнес-процесс состоит из нескольких шагов, которые должны быть разделены между членами группы для достижения общей цели,

программное обеспечение для автоматизации бизнес-процессов должно включать возможность представления этих шагов в графическом виде. Это даёт сотрудникам хорошее представление о том, что и когда нужно сделать и как процесс можно оптимизировать, а также упрощает организацию работы. Система автоматизации бизнес-процессов также должна быть проста в использовании, что обеспечивается благодаря интуитивному интерфейсу, адаптируемости и удобству доступа. Как небольшие, так и крупные компании всё чаще отдают предпочтение облачным системам для автоматизации бизнес-процессов и обеспечивают сотрудникам доступ к программному обеспечению из любой точки мира без привязки к конкретным десктопным компьютерам и мобильным устройствам. Также система автоматизации бизнес-процессов должна своевременно сообщать всем заинтересованным стейкхолдерам об изменении статуса протекающего процесса, а также рассылать приятные напоминания о том, что было сделано и что еще нужно сделать. Но оповещения ничего не значат, если нет отчетов.

Система автоматизации бизнес-процессов должна быть способна отразить прогресс бизнеса или компании в графиках и статистике и представить руководителям отделений организованный и точный обзор того, над чем подчинённые в настоящее время работают. Эти две комбинированные функции помогут руководителям всех уровней контролировать, что происходит, и вовремя принимать необходимые меры. Параллельные рабочие процессы – отличный способ, которым ваше программное обеспечение может показать прогресс, который делается, потому что он ломается с традиционной идеей, что вещи делаются на первом шаге, а затем еще один. В большинстве случаев люди, которые хотят достичь общей цели, выполняют некоторые задачи одновременно. Поэтому важна возможность построения и запуска нескольких параллельных бизнес-процессов, которые запускают друг друга при определённых заданных условиях. Но несмотря на то, что существуют общие цели, не вся информация, которая обрабатывается, предназначена для совместного использования со всеми сотрудниками. Именно поэтому контроль доступа на основе ролей

является одной из важнейших функций при выборе системы для автоматизации бизнес-процессов. Хорошая BPM-система должна обеспечивать возможность в любое время и быстро изменять уровни доступа для конкретных людей или групп, если это необходимо. Также немаловажна возможность интеграции автоматизированных бизнес-процессов с другими ИТ-системами, используемыми в компании, для ещё более эффективного управления бизнесом.

Для визуализации бизнес-процессов широко применяются разные виды UML диаграмм, а также BPMN. BPMN описывает условные обозначения и их описание в XML для отображения бизнес-процессов в виде диаграмм бизнес-процессов. При этом нотация языка такова, что понять суть происходящего на диаграмме за счет использования простых визуальных форм способен не только технический специалист, но и рядовой сотрудник компании.

Моделирование в BPMN осуществляется посредством диаграмм с небольшим числом графических элементов. Выделяют четыре основные категории элементов:

1) Объекты потока управления: события, действия и логические операторы (развилки)

Объекты потока управления разделяются на три основных типа: события (events), действия (activities) и логические операторы (gateways):

a) Event — это событие, которое произошло в описании процесса. Эти события могут быть начальными, конечными или промежуточными (рис.2.1).






























| | Начальные | Промежуточные | | Завершающие |
|-------------|---|---|--|---|
| | | Обработка | Генерация | |
| Простое |  |  | |  |
| Сообщение |  |  |  |  |
| Таймер |  |  | | |
| Ошибка | |  | |  |
| Отмена | |  | |  |
| Компенсация | |  |  |  |
| Условие |  |  | | |
| Сигнал |  |  |  |  |
| Составное |  |  |  |  |
| Ссылка | |  |  | |
| Останов | | | |  |

Рис.2.1 – Виды событий (events)

- Простые события (plain events) это нетипизированные события, использующиеся, чаще всего, для того, чтобы показать начало или окончание процесса.
- События-сообщения (message events) показывают получение и отправку сообщений в ходе выполнения процесса.

- События-таймеры (timer events) моделируют события, регулярно происходящие во времени. Также позволяют моделировать моменты времени, периоды и тайм-ауты.
- События-ошибки (error events) позволяют смоделировать генерацию и обработку ошибок в процессе. Ошибки могут иметь различные типы.
- События-отмены (cancel events) инициируют или реагируют на отмену транзакции.
- События-компенсации (compensation events) инициируют компенсацию или выполняют действия по компенсации.
- События-условия (conditional events) позволяют интегрировать бизнес правила в процесс.
- События-сигналы (signal events) рассылают и принимают сигналы между несколькими процессами. Один сигнал может обрабатываться несколькими получателями. Таким образом, события-сигналы позволяют реализовать широковещательную рассылку сообщений.
- Составные события (multiple events) моделирует генерацию и моделирование одного события из множества.
- События-ссылки (link events) используются как межстраничные соединения. Пара соответствующих ссылок эквивалентна потоку управления.
- События-остановы (terminate events) приводят к немедленному завершению всего бизнес процесса (во всей диаграмме).

б) Activity – это те действия (задачи), которые необходимо выполнить на определенном этапе бизнес-процесса. При моделировании их обычно обозначают прямоугольниками (рис.2.2), что соответствует сути действия.

Действия могут быть элементарными, т. е. неделимыми на некоторые более простые действия, и неэлементарными, т. е. такими, которые при детализации распадаются на последовательность определенных более простых действий.

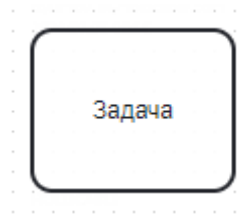


Рис.2.2 – Задача в BPMN

Задача (рис.2.2) – единица работы. Если задача помечена символом + (рис.2.3), то задача является подпроцессом и может быть детализирована.

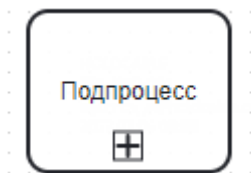


Рис.2.3 – Подпроцесс в BPMN

Транзакция (рис.2.3) – набор логически связанных действий. Для транзакции может быть определен протокол выполнения.

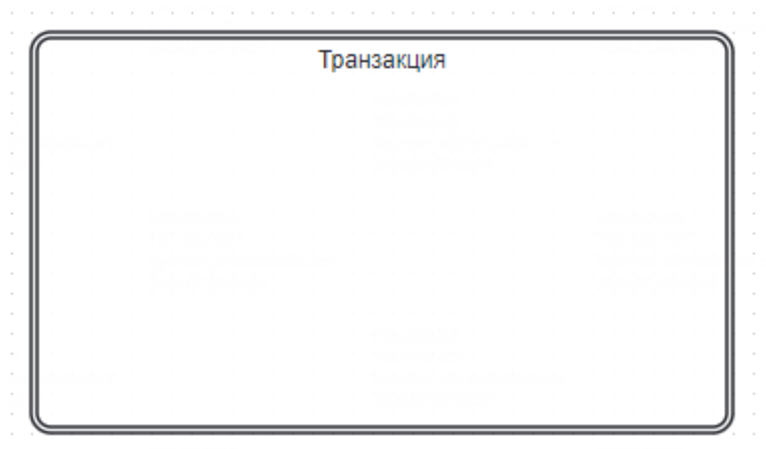


Рис.2.4 – Транзакция в BPMN

Событийный подпроцесс (рис.2.5) помещается внутри другого процесса. Он начинает выполняться, если инициируется его начальное событие. Событийный подпроцесс может прерывать родительский подпроцесс или выполняться параллельно с ним.

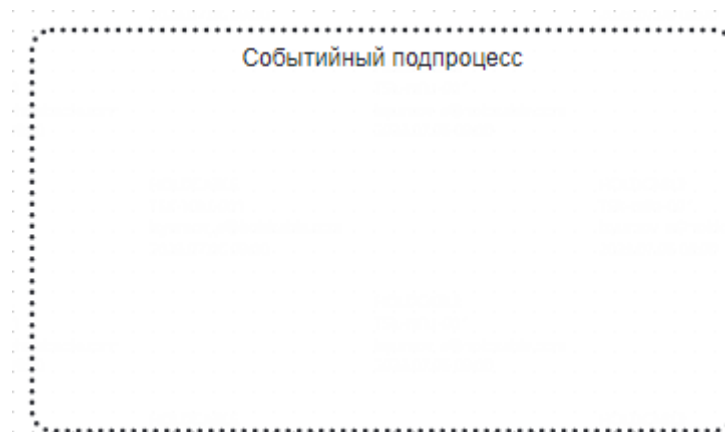


Рис.2.5 – Событийный подпроцесс в BPMN

Вызывающее действие (рис.2.6) является точкой входа для глобально определенного подпроцесса, который повторно используется в данном процессе.

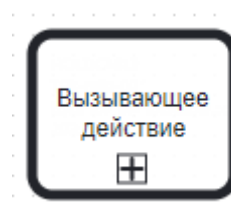


Рис.2.6 – Вызывающее действие в BPMN

в) GATEWAY — это управляющий узел, который появляется при условном разветвлении бизнес-процесса. Графически изображается в виде ромба.

Шлюзы нужны и в тех случаях, когда процедура зависит от определенных факторов. Например, при работе с покупателями шлюз появляется на этапе, когда клиент принимает решение о покупке — «да или нет». При положительном решении необходимо совершить покупку, при отрицательном - выяснить возможные причины отказа, работать с «отказом» и т.д.

Оператор исключающего ИЛИ (рис.2.7), управляемый данными. При ветвлении направляет поток лишь по одной из исходящих ветвей. При синхронизации потоков оператор ожидает завершения одной входящей ветви и активирует исходящий поток управления.



Рис.2.7 – Оператор исключающего ИЛИ в BPMN

Оператор И (рис.2.8). При разделении на параллельные потоки все ветви активируются одновременно. При синхронизации параллельных ветвей оператор ждет завершения всех входящих ветвей и затем активирует исходящий поток.



Рис.2.8 – Оператор И в BPMN

Оператор ИЛИ (рис.2.9). При ветвлении активируется одна или более ветвей. При слиянии все выполняющиеся входящие ветви должны быть завершены.



Рис.2.9 – Оператор ИЛИ в BPMN

Оператор триггер исключающего ИЛИ (рис.2.10). Активируется при условии активации управляющего события. Направляет поток управления лишь по той исходящей ветви, на которой первой произошло событие. После оператора данного типа могут следовать только события или действия-обработчики сообщений.



Рис.2.10 – Оператор триггер исключающего ИЛИ в BPMN

2) Соединяющие объекты: поток управления, поток сообщений и ассоциации

Поток управления изображается сплошной линией (рис.2.11), оканчивающейся закрашенной стрелкой. Поток управления задаёт порядок выполнения действий. Если линия потока управления перечеркнута диагональной чертой со стороны узла, из которого она исходит, то она обозначает поток, выполняемый по умолчанию.

Условный поток обозначается маленьким ромбом в начале стрелки соответствующего потока управления. Содержит условие, которое определяет, будет активирован данный поток или нет.



Рис.2.11 – Типы потоков управления в BPMN

Поток сообщений (рис.2.12) изображается штриховой линией, оканчивающейся открытой стрелкой. Поток сообщений показывает, какими сообщениями обмениваются участники.



Рис.2.12 – Поток сообщений в BPMN

Ассоциации (рис.2.13) изображаются пунктирной линией, заканчивающейся стрелкой. Ассоциации используются для ассоциирования артефактов (данных или текстовых аннотаций) с объектами потока управления

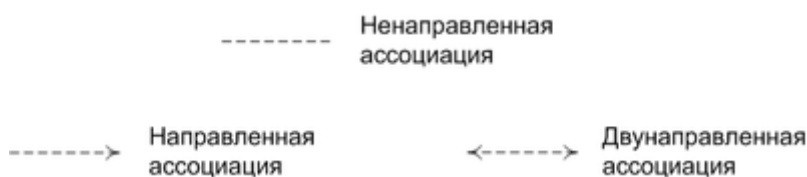


Рис.2.13 – Типы ассоциаций в BPMN

3) Роли: пулы и дорожки

Пул – это объект, описывающий один процесс на диаграмме. Его может не быть на карте, но он всегда есть. На одной диаграмме может быть несколько Пулов. Пул может быть расширен для просмотра деталей.

Пул также может содержать так называемые «дорожки» (рис.2.14). Они нужны для того, чтобы указать участников процессов, которые скрыты в пуле. Например, в процессе работы с клиентами участвует менеджер по продажам, начальник отдела продаж, возможно, бухгалтер или кассир.



Рис.2.14 – Пул и дорожки в BPMN

Пулы (участники) и дорожки отражают распределение обязанностей. Пул или дорожка обозначает организацию, роль или систему. Дорожки позволяют иерархически делить пулы и другие дорожки.

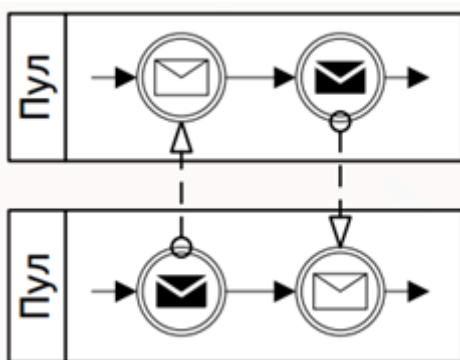


Рис.2.15 – Иерархия пулов в BPMN

Порядок обмена сообщениями может быть задан при помощи потока сообщений и потока управления.

4) Артефакты: данные, группы и текстовые аннотации

Объект данных (рис.2.15) — это элемент, указывающий, какие данные и документы необходимы для начала действия или каковы результаты завершенного действия.

Объект данных может быть сгенерированным заказом. Для менеджера это будет результат действий, а для склада, принимающего заказ, начало действия (сбор товара и отгрузка).



Рис.2.15 – Данные в BPMN

Тип элемента **Сообщение** (рис.2.16) необходим для отображения отношений между двумя участниками процесса.



Рис.2.16 – Сообщение в BPMN

Это может быть электронное письмо, сообщения внутри системы совместной работы (например, в Google Docs), переписка в любом из мессенджеров, которыми пользуются участники процесса, общение на сайте компании, СМС-сообщения и т.д.

Под **артефактами** (рис.2.17) в BPMN понимаются объекты, не являющиеся действиями и не имеющие прямого отношения к действиям. Это могут быть любые документы, данные, информация, не влияющие непосредственно на выполнение процесса.

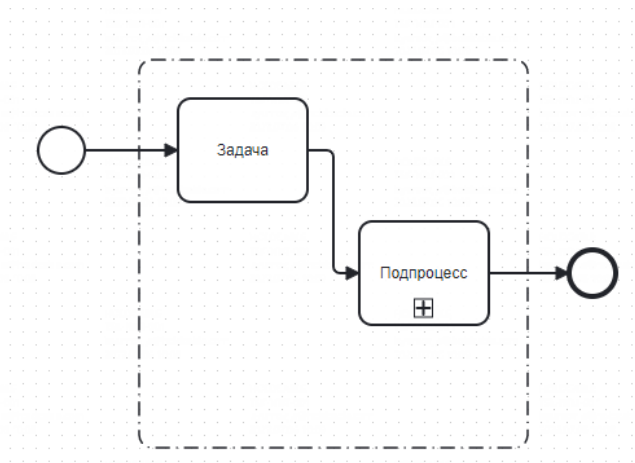


Рис.2.17 – Пример группы объектов с двумя задачами, стартам и финалом

Существует два типа артефактов:

- Группа объектов;
- Текстовая аннотация.

Группа объектов — это еще один способ объединить несколько элементов под общим символом, чтобы сэкономить место на диаграмме и упростить ее чтение. Здесь различные виды деятельности собраны под одним общим названием. Группа объектов также всегда может быть подробно рассмотрена. Группа (рис.2.18) выглядит как прямоугольник со скругленными углами, выполненный пунктирной линией с точками.

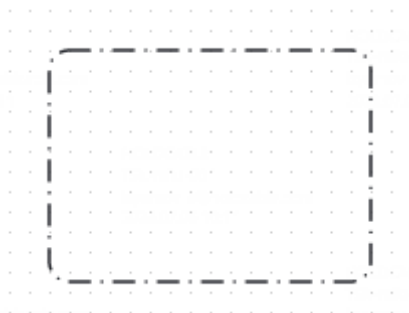


Рис.2.18 – Группа объектов в BPMN

Текстовые аннотации (рис.2.19) используются для различных уточнений к диаграмме. Это могут быть комментарии, пояснения, другая информация, которая повысит читабельность схемы. Аннотации представляют собой

незамкнутый прямоугольник, выполненный сплошной линией, от которого к объекту аннотации ведет линия, состоящая из точек [2].



Рис.2.19 – Текстовые аннотации в BPMN

2.2 Задание

Выполнить лабораторную работу. Выполненная лабораторная работа оформляется в виде отчета (формат документа *.pdf) и прикрепляется к соответствующему элементу электронного курса.

2.3 Порядок выполнения работы

Используя любое online средство по работе с BPMN диаграммами (например, Lucid.app, bpmn.io и т.д.) создать BPMN диаграмму проведения online-занятия с использованием программы Zoom. Предусмотреть не менее 8 шагов.

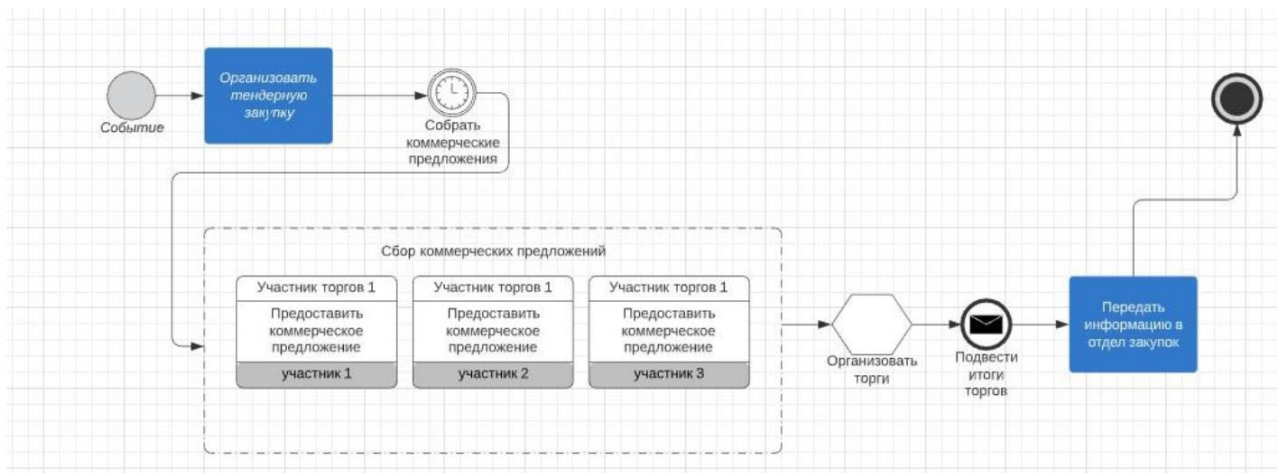


Рис.2.20 - Пример BPMN диаграммы проведения тендерной закупки

2.4 Содержание отчета

1. Титульный лист, оформленный в соответствии с Образовательным стандартом ТУСУР (<https://regulations.tusur.ru/documents/70>).
2. На следующем листе сверху цель данной лабораторной работы.
3. Ход выполнения работы (содержащий пошаговое описание выполненных действий и скриншоты полученных результатов).
4. Выводы.

Лабораторная работа №2 «Основы работы в 1С. Справочники»

Цель работы: знакомство с основами конфигурирования в 1С.

Темы для предварительного изучения: 1С режимы работы. Базовый синтаксис 1С. Переменные. Функции. Процедуры. Управляющие конструкции. Контекст. Справочники.

3.1 Основы работы в 1С

В плане обработки информации СУБД обладают значительно большими возможностями, чем электронные таблицы. В отличие от случаев, когда данные представлены в виде простых таблиц (т.е. в машинном виде реализованы в табличном редакторе), данные в БД логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе. [3,4]

Платформа 1С является по сути представителем СУБД, но одновременно и мощным средством визуального программирования.

Язык 1С весьма прост в изучении по сравнению с любым другим языком программирования. На территории России для многих большое значение имеет то, что можно писать код прямо на русском языке. «Можно» – значит не обязательно — можно писать и на английском, все операции и операторы имеют английский синоним.

Программирование – это написание программного кода на программном языке. Программный код – это последовательность команд (слов), которые поддерживает этот язык программирования. Команды делятся на:

- Переменные
- Операторы
- Функции и процедуры.

Рассмотрим их в первом приближении.

Переменные

Есть цифра 12. Это значение. Сама по себе эта цифра нигде не хранится. Подули и ее уже нет. Она должна где-то храниться. Переменные – это место хранения значений. Переменная может называться любым словом.

Например:

```
Чтото = 12;  
Чтото = Чтото + 10;  
Сообщить(Чтото); //будет выведено «22»
```

Переменная, которая является частью объекта 1С (например поле справочника) и хранится в базе данных – называется ***реквизит***.

В 1С переменные не типизированы, это значит, что одной и той же переменной сначала можно назначить значение одного типа (например цифру), а потом другого типа (например строку) и ошибки не будет. Однако реквизиты объектов – типизированы. Но проверки на типы при этом не происходит.

Мы можем попытаться назначить реквизиту значение не того типа. 1С попытается его преобразовать (например из цифры в строку), но если не получится, то значение реквизита останется пустым.

Длина строки в переменной – неограниченная. Длина строки реквизита – задается точно, но можно задать неограниченную (поставив длину строки 0). По буквам к строке обращаться нельзя. Но можно искать или выделять части (Найти(), Лев(), Прав(), Сред()).

Модули

Любая программа состоит из программного кода, то есть собственно из написанных на каком-либо языке последовательности действий, которые

должны быть выполнены. Однако эта самая программа должна быть где-то написана, то есть где-то находиться. В большинстве случаев код программы пишется в обычных текстовых файлах. Разница только в том, что расширение в них не .txt, а .cpp или .php. Что такое Модуль 1С? Конечно код 1С можно было бы тоже написать в каком-нибудь текстовом файле. Однако есть понятие **Конфигурация 1С** – которое включает в себя не только список настроек, шаблонов форм и прочего, но и программный код 1С. Поэтому код 1С хранится в конфигурации. Каждый объект 1С содержит в себе вложенные объекты, например справочник имеет несколько форм. Каждый объект 1С, включая некоторые вложенные, имеет свой **Модуль** – некий текстовый файл, который содержит программный код. Также есть независимые от объектов модули, в которых может быть написан программный код, независимый от конкретного объекта. Таким образом в 1С нет «единой» программы. Есть набор модулей для написания программного кода для каждого объекта конфигурации 1С.

Клиент и сервер 1С

В серверном варианте 1С разные части программы выполняются на клиентской машине или на сервере. Зачем это нужно мы обсудим потом, для начала можно взять хотя бы производительность клиентского компьютера и сервера. У общих модулей (Общие / Общие модули) есть свойства. Нажмите правой кнопкой мыши и далее – Свойства. Галочками можно отметить где будет доступен этот модуль. Также один модуль может выглядеть по-разному если запущен на сервере или клиенте. Для этого есть специальные директивы, указываемые прямо в коде программы: #Если Клиент и #Если Сервер.

Клиент и **Сервер** видят только те части модуля, которые находятся между этими директивами. Если директив нет – модуль виден весь. В версии 1С 8.2 для тонкого клиента можно указывать для каждого метода и обработчика где он будет выполняться. Для этого непосредственно перед функцией или процедурой указывается &НаСервере и &НаКлиенте. При этом клиентские функции могут вызывать серверные, но не наоборот.

Сообщение между модулями 1С

Каждый модуль существует сам по себе и в него нет точек входа извне — из других модулей. Если мы хотим создать метод или переменную, видимую из другого модуля, она помечается словом ЭКСПОРТ. Однако большое количество модулей и правил клиент-сервера порождает некоторое количество проблем у программиста с использованием модуля извне. Где-то все просто, а где-то можно тупить пол часа — почему написано ЭКСПОРТ, но вызов оканчивается ошибкой «метод не найден..».

Контекст

Под ***контекстом*** подразумевается весь тот набор функций, процедур и переменных, доступных на определенной строке выполнения программы, с теми значениями переменных, которые они уже имеют на данном этапе выполнения.

Строки программы

Строки заканчиваются на точку с запятой «;». Комментарий начинается с «//» — то есть эта строчка или часть строчки не будет выполняться и будет пропущена. Значение строк нужно указывать в кавычках «Значение». Если нужно указать в значении кавычку, то она удваивается — «Значение ««а»»!». Значение строки может быть с включением переноса строки, тогда в начале следующей строки должен стоять «\», например:

«Это начало | и продолжение значения»

Если нужно указать специальные символы в значении строки, для этого есть специальный объект: «Символы.», например:

«Это начало» + Символы.ПС + «и продолжение значения» //ПС – перенос строки

Операторы

Это конструкции языка, которые заложены в нем самом и позволяют создавать простейшую последовательность выполнения программы.

Условие

Определенные строки кода будут выполнены если будет выполнено условие.

```
Чтото = 12;
```

```
Если Чтото < 100 или Чтото > 200 Тогда Сообщить («Условие выполнено»);
```

```
ИначеЕсли Чтото > 200 Тогда Сообщить («Условие не выполнено»);
```



```
Иначе Сообщить («Что-то еще»);  
КонецЕсли;
```

Цикл

Определенные строки кода будут выполнены указанное количество раз.

```
//считаем по-одному  
Для Чтото = 1 по 20 Цикл  
    Сообщить (Чтото) ;
```

```
КонецЦикла;
```

```
//считаем по-другому
```

```
Чтото = 1;
```

```
Пока Чтото < 20 Цикл
```

```
    Чтото = Чтото + 1;
```

```
    Сообщить (Чтото) ;
```

```
КонецЦикла;
```

```
//если у нас список значений, то можно обойти каждое из его значений
```

```
Чтото = Новый СписокЗначений;
```

```
Для каждого ЗначениеСписка из Чтото Цикл
```

```
    Сообщить (ЗначениеСписка.Значение) ;
```

```
КонецЦикла; //обратите внимание, что «ЗначениеСписка» - это  
переменная, она может называться как угодно
```

Для того, чтоб прервать цикл, можно внутри написать команду «Прервать», для того чтобы прервать только одну итерацию цикла – команду «Продолжить».

Работа с объектами

Все объекты, с которыми мы работаем в языке – являются полноценными объектами, то есть могут иметь свои данные и свои методы. Объекты языка создаются с помощью команды «Новый», например: Чтото = Новый Массив(); Чтото = Новый СписокЗначений(); И т.п. Объекты 1С нельзя создать – доступ к ним можно получить с помощью так называемых менеджеров, названных соответственно по веткам конфигурации: Справочники, Документы и т.п. Методы менеджера можно расширять. Например:
Справочники.Номенклатура.СоздатьЭлемент();
Документы.Накладная.НайтиПоНомеру(«...»);

Свои объекты программировать нельзя, как и наследовать или переопределять типовые. Ошибки Программист не всегда может догадаться, будет ли ошибка при выполнении определенной строчки кода. Например если написать в коде внешней обработки: Справочники.Номенклатура.НайтиПоКоду(«...») и открыть эту обработку в базе, где нет справочника с названием Номенклатура, то будет ошибка. Чтобы этого не произошло, можно использовать следующую конструкцию:

```
Попытка
    Справочники.Номенклатура.НайтиПоКоду («...»);
Исключение
    Чтото = ОписаниеОшибки();
    Сообщить («Ой-ой-ой»);
    Сообщить (Чтото);
КонецПопытки;
```

Если в момент выполнения строчек программы между Попытка и Исключение возникнет ошибка, то выполнение будет прервано и перейдет на строчки между Исключение и КонецПопытки. Если не будет ошибки – эти строчки не будут выполнены. Функция ОписаниеОшибки() работает только между Исключение и КонецПопытки и возвращает описание произошедшей ошибки.

Функции и Процедуры

Функции и Процедуры – это способ взять несколько строчек кода и назвать их каким-то словом, как переменную. Когда мы потом напишем ее название в другом месте – будет вызвана эта функция (то есть выполнены эти строки кода). Мы неоднократно обсуждали методы объектов и обработчики событий. Функции и процедуры и являются этими методами и обработчиками.

```
//Программируем
Процедура КакоетоДействие()
    Сообщить («Чтото»);
КонецПроцедуры

//Вызываем
КакоетоДействие();
```

Функция отличается от процедуры тем, что она может вернуть значение, которое было вычислено в результате выполнения этих строк когда. Например:

```
Функция Посчитать (Переменная1, Переменная2)
    Возврат Переменная1 + Переменная2;
КонецФункции
//Вызываем
Чтото = Посчитать (12, 20);
Сообщить (Чтото); //будет сообщение «32»
```

Справочники

Справочники являются основным механизмом хранения условно-постоянной информации. Справочники часто используются в тех случаях, когда необходимо исключить неоднозначный ввод информации. Каждый справочник представляет собой список однородных объектов: сотрудников, организаций, товаров и т.д. Такие объекты называются элементами справочника.

Замечание. Для упрощения понимания можно считать, что справочник — это обычная таблица с заданными колонками. Но в 1С:Предприятии 8. есть возможность создания иерархических справочников и табличных частей у справочников. Отсюда можно сделать вывод, что справочник — это не просто таблица, а достаточно сложный механизм хранения данных.*

На этапе разработки можно описать, какими свойствами обладает справочник. К настраиваемым свойствам относятся длина и тип кода, количество уровней иерархии, поддержка уникальности кодов, перечень реквизитов справочника и его табличных частей, а также другие параметры.

Структура справочника

Структура справочника определяет состав хранимой информации и настраивается в Конфигураторе. Под структурой понимаются в первую очередь состав реквизитов и табличных частей справочника. В более широком смысле в понятие структуры справочника включают тип и длину кода, длину наименования, количество уровней иерархии и т.д.

Во всех справочниках есть системные реквизиты Код и Наименование (если только у них не установлена нулевая длина). Действительно, практически

любой объект из реальной жизни характеризуется этими свойствами. Например, для сотрудников кодом может выступать табельный номер, а наименованием — фамилия, имя и отчество (ФИО). Код элемента справочника может быть как числовым, так и строковым. Система 1С:Предприятие 8.* предоставляет широкие возможности по работе с кодами элементов справочника: автоматическое присвоение кодов новым элементам, контроль уникальности кода и т.д.

Помимо кода и наименования, в справочниках 1С:Предприятия 8.* может храниться любая дополнительная информация об объектах предметной области, для чего в Конфигураторе настраивается список реквизитов и табличных частей. Используя механизм справочников легко организовать, например, картотеку сотрудников.

Почти наверняка справочник Сотрудники, кроме встроенных реквизитов Код и Наименование, будет иметь какие-то дополнительные реквизиты, например, Должность, Оклад, Адрес, ДатаРождения и т.д. Для каждого реквизита справочника нужно указать его тип данных, например, «число», «строка», «дата», «булево». Это примитивные типы данных, но можно указать и ссылочные типы данных. Например, реквизит Должность имеет тип данных «СправочникСсылка.Должности». В этом случае, значение реквизита будет выбираться из справочника Должности. Так реализуется простейшая связь между справочниками и другими объектами, когда значения реквизита объекта выбираются из справочника.

Предопределенные элементы

Еще на этапе разработки конфигурации в справочник могут быть введены предопределенные элементы и группы справочника. Эти элементы имеют имена, по которым к ним производится обращение из программы.

Предопределенные элементы справочника имеет смысл создавать, если сама конфигурация ориентирована на их использование. Например, иерархический справочник Сотрудники может иметь две предопределенные группы: Работающие и Уволенные. Эти группы будут использованы в приказе о

приеме на работу и приказе об увольнении. При приеме сотрудник добавляется в группу Работающие, а при увольнении переносится в группу Уволенные.

Подчиненные справочники

Между справочниками может быть установлено отношение подчиненности. В этом случае каждый элемент подчиненного справочника будет связан с одним из элементов справочника-владельца. Часто можно сказать, что элементы одного справочника принадлежат элементам другого. Например, в системе может быть справочник РасчетныеСчета, тогда его можно сделать подчиненным справочнику Контрагенты. Это означает, что «Контрагент владеет расчетными счетами, а расчетные счета принадлежат контрагенту» В 1С:Предприятии 8.* у справочника может быть несколько владельцев, то есть справочник может быть подчинен сразу нескольким справочникам или другим объектам. Но каждый элемент подчиненного справочника имеет одного и только одного владельца, который хранится в одном из объектов-владельцев.

3.2 Задание

Выполнить лабораторную работу. Выполненная лабораторная работа оформляется в виде отчета (формат документа *.pdf) и прикрепляется к соответствующему элементу электронного курса.

3.3 Порядок выполнения работы

- 1) Создадим пустую конфигурацию (рис.3.1). В окне запуска 1С нажмем «Добавить». Выберем вариант с созданием новой базы.

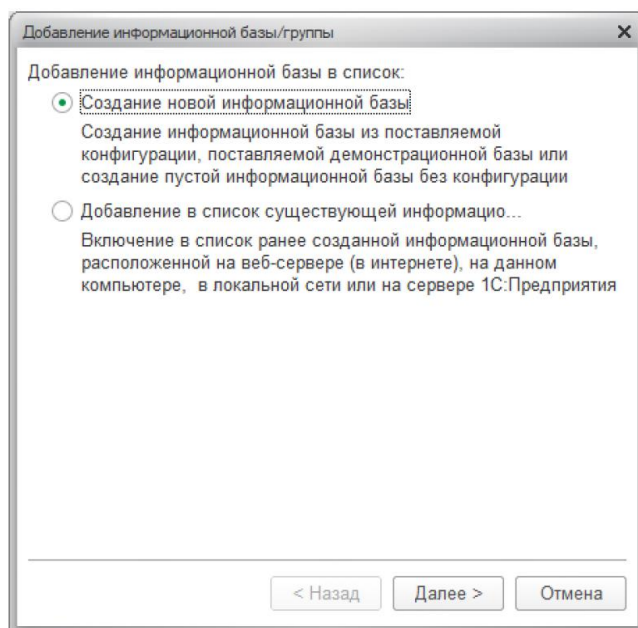


Рис.3.1 – Добавление информационной базы. Шаг 1

2) Выберем вариант «база без конфигурации» (рис.3.2).

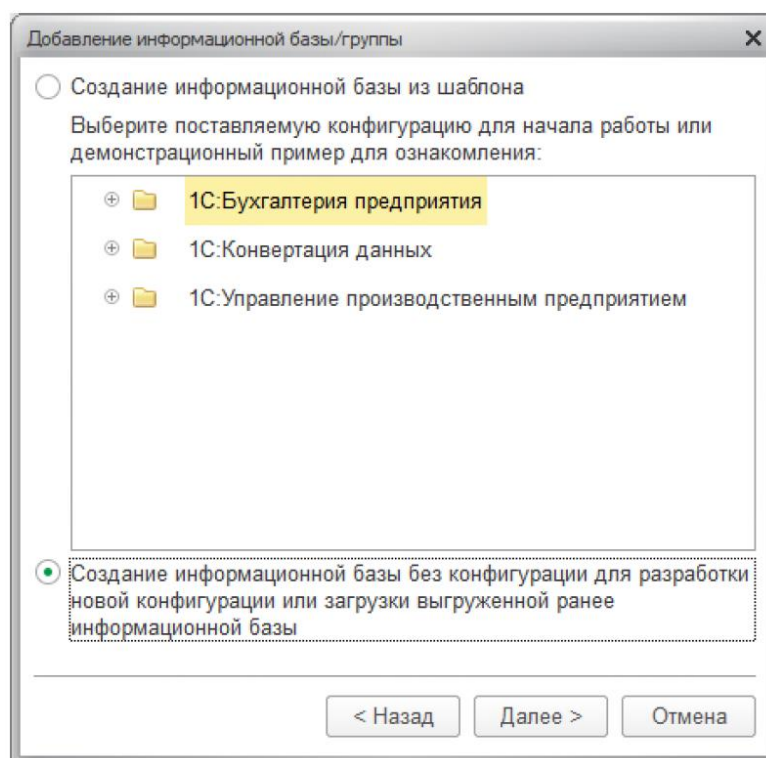


Рис.3.2 – Добавление информационной базы. Шаг 2

3) Введем имя и укажем расположение (рис.3.3-3.5).

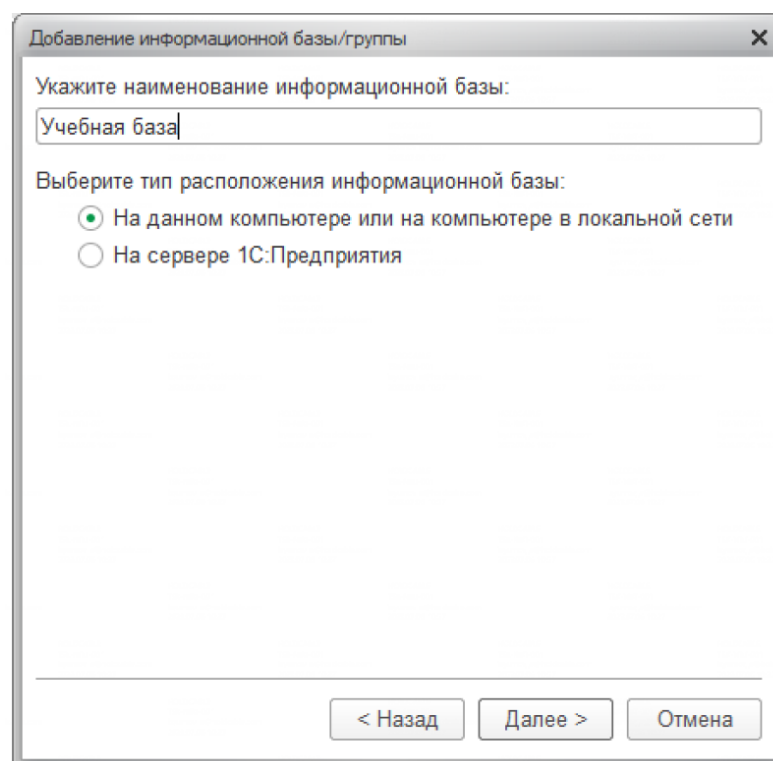


Рис.3.3 – Добавление информационной базы. Шаг 3

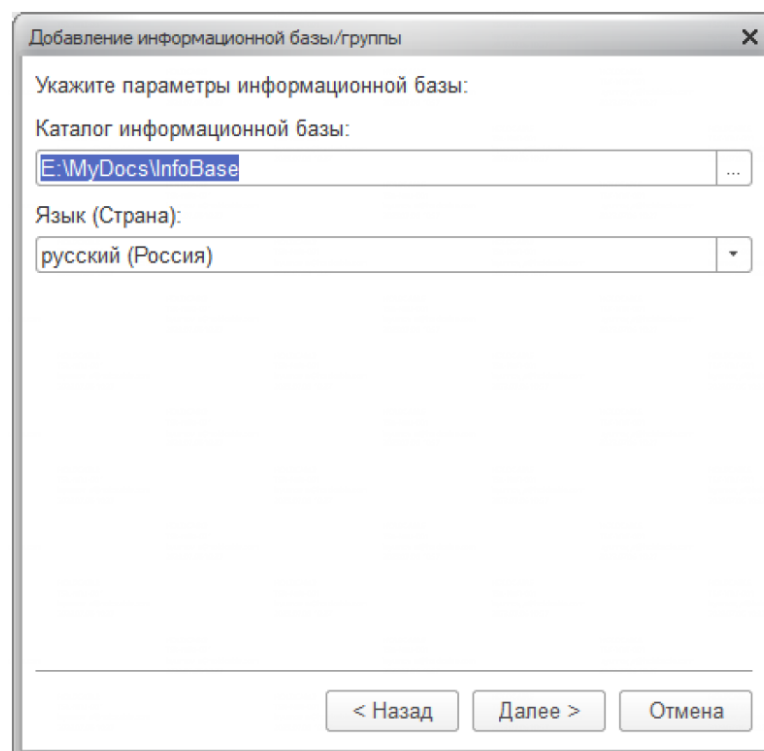


Рис.3.4 – Добавление информационной базы. Шаг 4

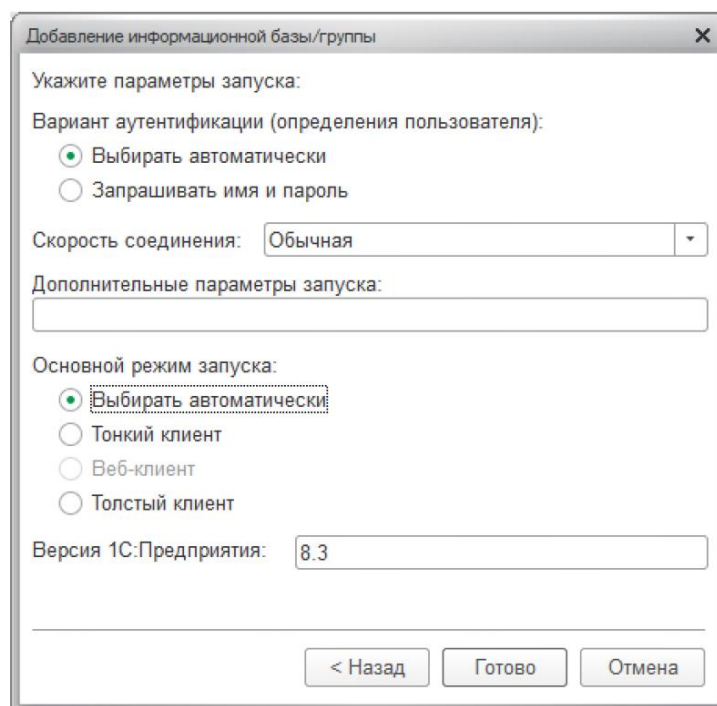


Рис.3.5 – Добавление информационной базы. Шаг 5

- 4) Далее в Конфигураторе нажмем Конфигурация-Открыть конфигурацию.
- 5) Создадим простой справочник «Должности» без дополнительных параметров. Должностью будет выступать ключевой реквизит Наименование (рис.3.6).

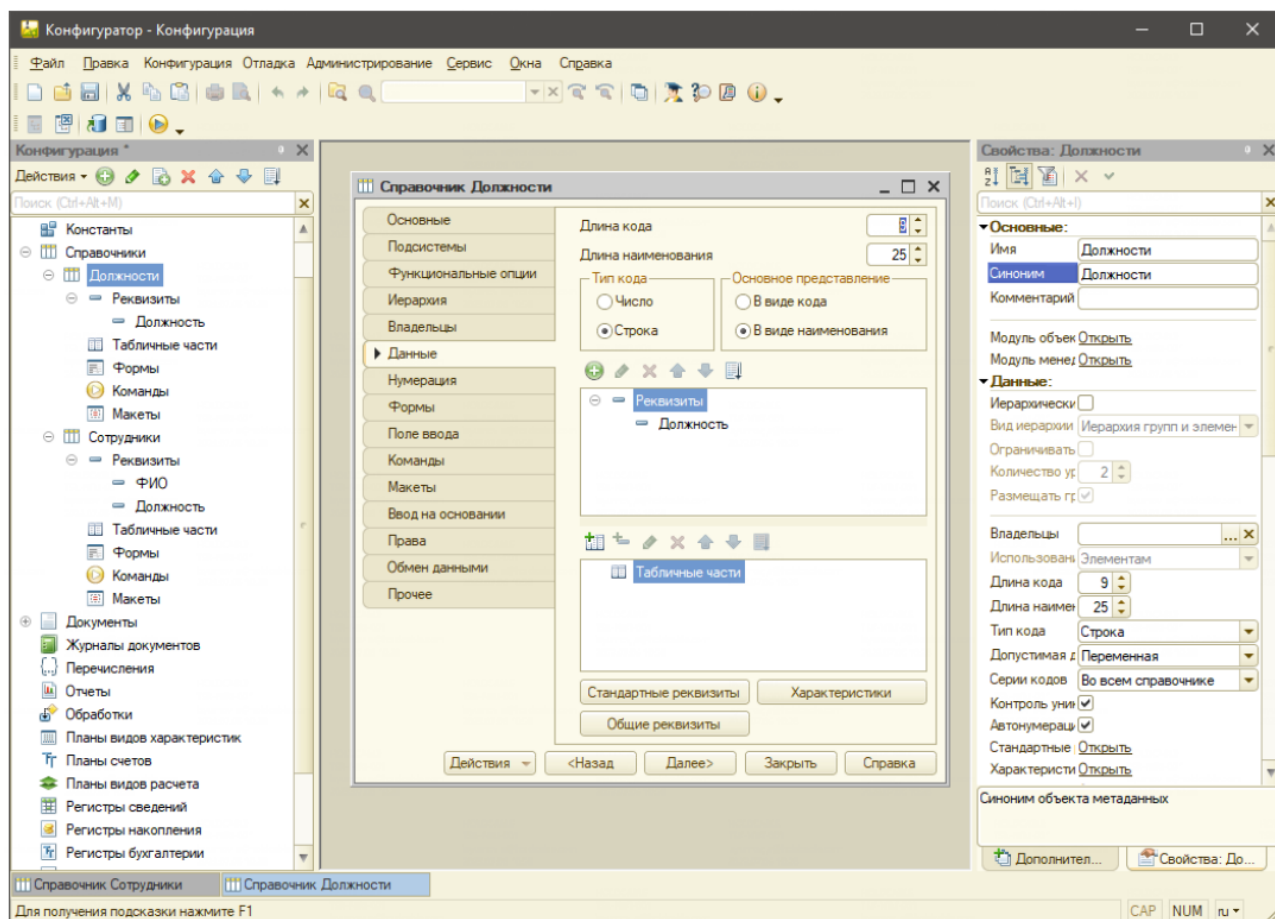


Рис.3.6 – Создание справочника «Должности»

б) Создадим справочник «Сотрудники» (рис.3.7). Справочник должен быть иерархическим.

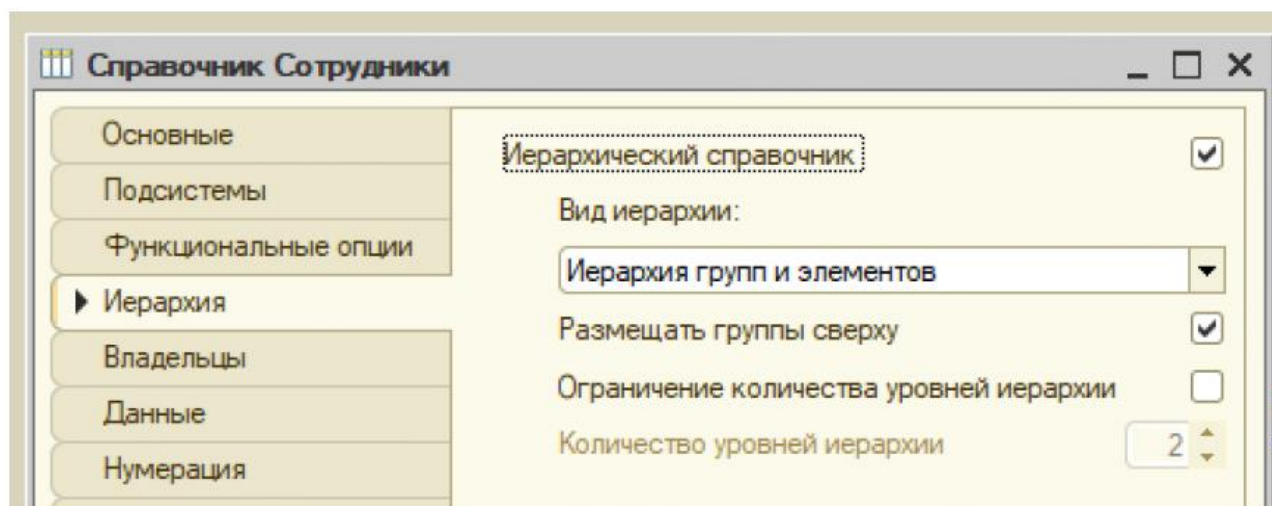


Рис.3.7 – Задание иерархичности справочника «Сотрудники»

Реквизиты справочника: - Должность (СправочникСсылка «Должности») - Оклад (число) (рис.3.8).

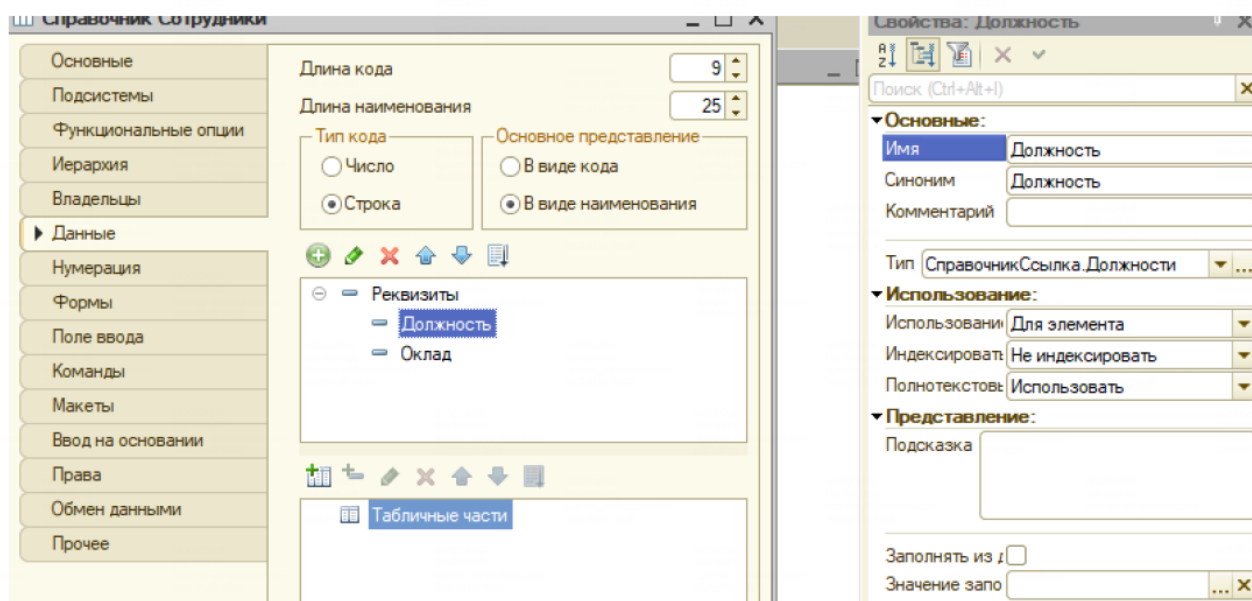


Рис.3.8 – Задание параметров реквизитов справочника

7) Создадим форму элемента справочника (рис.3.9).

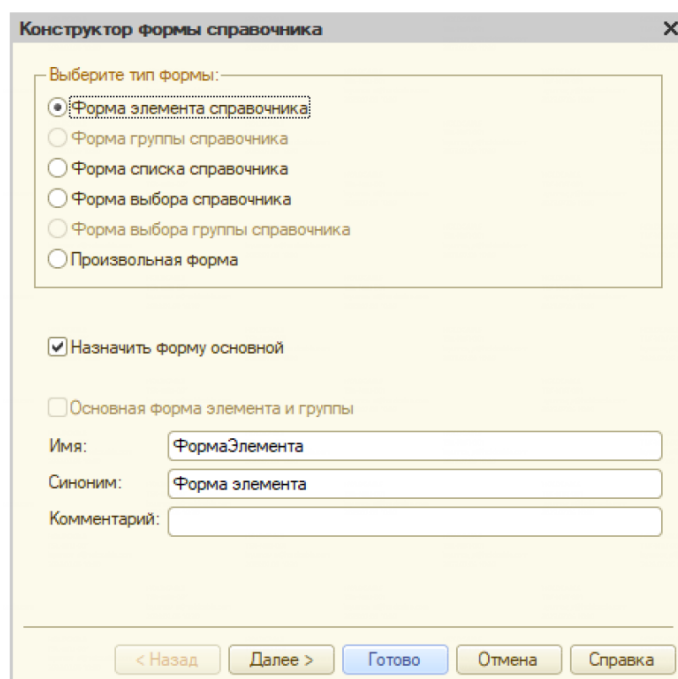


Рис.3.9 – Создание формы справочника

8) На форме элемента разместим дополнительную кнопку и свяжем ее с созданной командой формы «Сформировать» через признак «Действие» (рис.3.10).

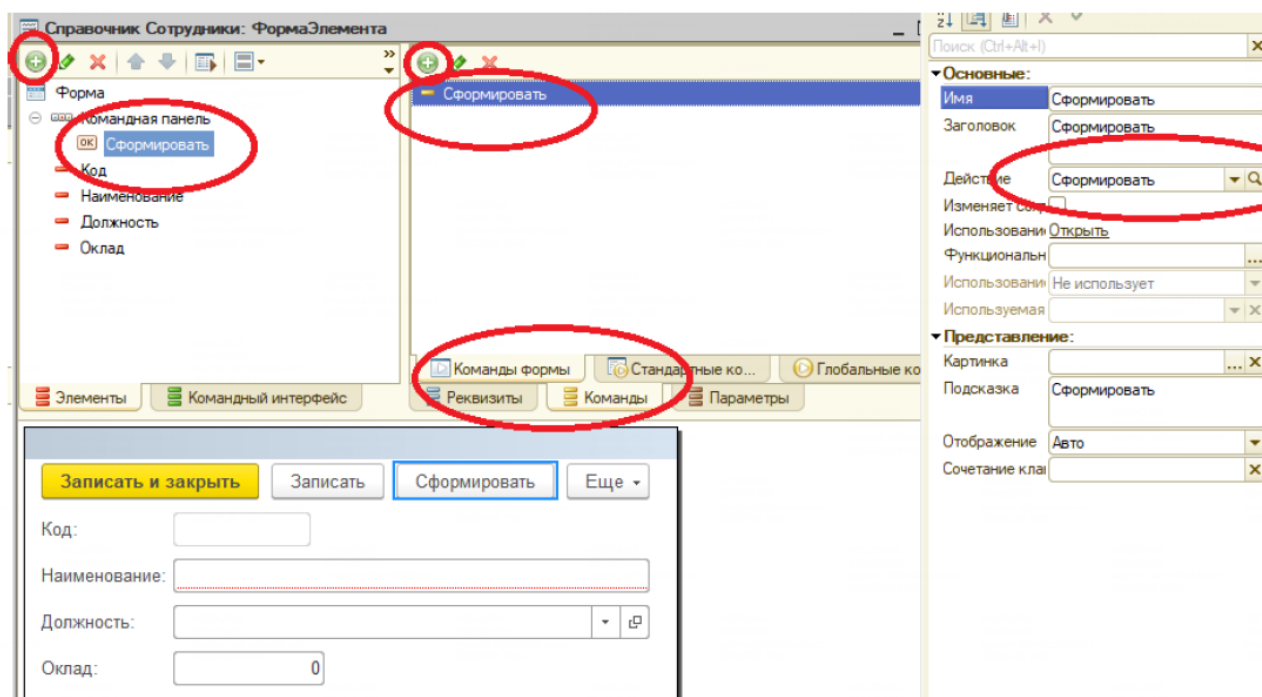


Рис.3.10 – Задание связей между элементами

При нажатии кнопки «Сформировать» в группу «Работающие» справочника «Сотрудники» добавляется новый элемент с данными, указанными в полях ввода

9) Сгенерируем команду для формы и зададим ее обработчик (рис.3.11)..

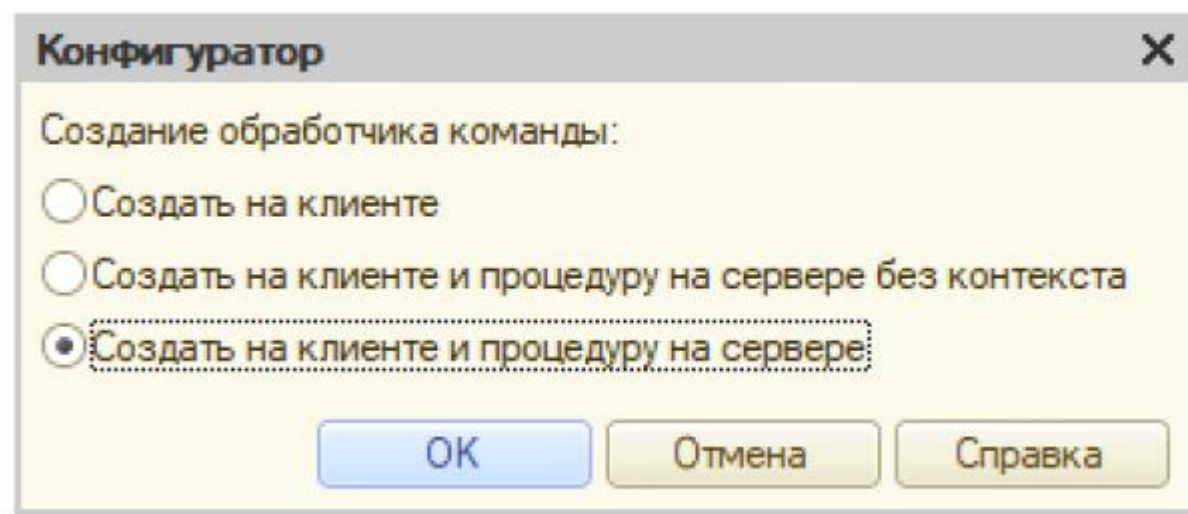


Рис.3.11 – Создание команды

В обработчике кнопки «Сформировать» запишем процедуру:

```
&НаКлиенте Процедура Сформировать (Команда)
Перем НовыйСотрудник;
НовыйСотрудник = Новый Структура ("Наименование, Должность, Оклад");
    НовыйСотрудник.Наименование          =          Объект.Наименование;
    Сообщить (Объект.Наименование) ;
    НовыйСотрудник.Должность = Объект.Должность;
    НовыйСотрудник.Оклад = Объект.Оклад;
    Если СоздатьНовогоСотрудника (НовыйСотрудник) = 0 Тогда
        Сообщить ("Не удалось создать нового сотрудника");
    КонецЕсли;
КонецПроцедуры

&НаСервере Функция СоздатьНовогоСотрудника (СтруктураСправочника)
Перем НовыйСотрудник, СпрПользователи;
    СпрСотр = Справочники.Сотрудники;
    НовыйСотрудник = СпрСотр.СоздатьЭлемент ();
    НовыйСотрудник.Наименование          =
    СтруктураСправочника.Наименование;
    НовыйСотрудник.Должность = СтруктураСправочника.Должность;
    НовыйСотрудник.Оклад = СтруктураСправочника.Оклад;
    СсылкаНаГруппу          =
    Справочники.Сотрудники.НайтиПоНаименованию ("Работающие") ;
    Если СсылкаНаГруппу.Пустая () Тогда
        НГ = Справочники.Сотрудники.СоздатьГруппу ();
        НГ.Наименование = "Работающие";
        НГ.Записать ();
        СсылкаНаГруппу = НГ.Ссылка;
    КонецЕсли;
    НовыйСотрудник.Родитель = СсылкаНаГруппу;
    Попытка НовыйСотрудник.Записать ();
        Возврат 1;
    Исключение
        Возврат 0;
    КонецПопытки;
КонецФункции
```

3.4 Содержание отчета

1. Титульный лист, оформленный в соответствии с Образовательным стандартом ТУСУР (<https://regulations.tusur.ru/documents/70>).
2. На следующем листе сверху цель данной лабораторной работы.
3. Ход выполнения работы (содержащий пошаговое описание выполненных действий и скриншоты полученных результатов). В документ вставить скриншоты созданных справочников, код обработчиков действий и вывод информационной панели.
4. Выводы.

Лабораторная работа №3 «Основы работы в 1С. Документы. Проведение»

Цель работы: знакомство с методами работы с документами в 1С и их проведением.

Темы для предварительного изучения: Документы. Оперативное и неоперативное проведение.

4.1 Основы работы с документами в 1С

Документ – одно из основных понятий системы 1С: Предприятие. При помощи документов организуется ввод в систему информации о совершаемых хозяйственных действиях. В большинстве своем документы, которые создаются в процессе настройки конфигурации задачи, являются электронными аналогами стандартных бумажных документов, однако, использование этого типа данных может выходить далеко за рамки простой фиксации информации о хозяйственных операциях.

Каждый документ содержит информацию о конкретной хозяйственной операции и характеризуется своим номером, датой и временем. Дата и время — наиболее важные характеристики документов, так как позволяют устанавливать строгую временную последовательность совершения операций. В конфигурации описывается только структура документа, конкретные экземпляры документов вводятся при работе с программой пользователем.

Например, созданный в конфигурации документ «Накладная» при работе с системой 1С: Предприятие позволит формировать накладные, которые будут иметь разное содержание, но одинаковый набор реквизитов, одинаковую логику поведения и так далее.

Конфигуратор позволяет описать структуру документа, организовать диалог для ввода информации в документ и описать алгоритм построения печатной формы документа.

Характерной особенностью объекта конфигурации Документ является то, что в процессе работы пользователь может самостоятельно создавать новые объекты этой структуры – новые документы.

Наиболее важными характеристиками документа, отличающими его от других объектов, является наличие у него номера и даты. В 1С:Предприятии 8.* значения типа «дата» включают в себя также и время.

Дата и время — наиболее важные характеристики документов. Все документы вне зависимости от вида образуют единую хронологическую последовательность. Фактически эта последовательность отражает последовательность событий — так, как они происходили реально. Внутри даты последовательность документов определяется их временем, при этом время документа является не столько средством отражения реального (астрономического) времени ввода документа, сколько средством, позволяющим четко упорядочить документы внутри одной даты. Если получится так, что два документа имеют одинаковую дату и время, то они все равно выстраиваются в последовательность.

Данные, вводимые в реквизиты документа, обычно содержат информацию о событии, например, в расходной накладной — информацию о том, с какого склада, каких товаров и сколько отгружено; в приказе о приеме на работу — информацию о сотруднике, оклад и другие сведения; в договоре с клиентом — условия договора, график плановых платежей и т.д. Кроме собственно записи, для документа весьма важным аспектом является его проведение. Именно при проведении документ обычно отражает зафиксированное им событие в регистрах 1С:Предприятия 8.*, например, в виде движений по регистру бухгалтерии и/или движений по регистру остатков товаров.

Оперативное и неоперативное проведение

Поведение документа отличается от поведения всех остальных объектов базы данных. Документ обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета. До тех пор, пока документ не проведен, состояние учета неизменно и

документ - не более чем черновик, заготовка. Как только документ будет проведен изменения, вносимые документом в учет, могут вступить в силу и состояние учета может быть изменено.

Поскольку документ вносит изменения в состояние учета, он всегда «привязан» к конкретному моменту времени. Это позволяет отражать в базе данных фактическую последовательность событий.

Следующим важным фактом, вытекающим из двух предыдущих, является то, что система 1С:Предприятие имеет механизмы, позволяющие отслеживать правильность состояния учета. Предположим, что мы изменили один из проведенных ранее документов и снова провели его «задним числом». В этом случае система 1С:Предприятие способна отследить, повлияют ли внесенные нами изменения на последующие проведенные документы, и если это так, система способна перепровести необходимые документы.

Факт проведения документа и необходимость поддержания актуальной последовательности документов на оси событий порождают два различных способа проведения документов: оперативное и неоперативное проведение.

С оперативным проведением документов связано понятие оперативной отметки времени. Оперативная отметка времени представляет собой значение типа Дата, которое формирует система. Оперативная отметка времени создается системой каждый раз при оперативном проведении документа. Ее значение формируется исходя из текущего времени и последней созданной оперативной отметки. Если последняя оперативная отметка меньше текущего времени, в качестве новой оперативной отметки принимается текущее время. Если последняя оперативная отметка равна или больше текущего времени, в качестве новой оперативной отметки принимается значение, на одну секунду большее, чем старая оперативная отметка времени. Таким образом, если у объекта конфигурации Документ установлено свойство оперативного проведения, последовательность действий системы будет следующей: при создании нового документа система будет устанавливать ему текущую дату и «нулевое» время. При проведении такого документа (с текущей датой), система установит в

качестве даты документа оперативную отметку времени. Если отменить проведение документа и затем провести его снова (не изменяя даты), система установит документу новую оперативную отметку времени. Если же попытаться перепровести документ, то будет выдан запрос о виде проведения (оперативное или нет). В случае оперативного проведения система установит новую оперативную отметку времени, а при неоперативном проведении время документа будет сохранено прежним. При попытке проведения (или перепроведения) оперативно проводимого документа с датой, отличающейся от текущей, будет выдано сообщение о том, что оперативное проведение невозможно и предложено провести документ неоперативно (т.е. с сохранением существующей даты и времени документа).

Структура документа

Под структурой документа в первую очередь понимаются реквизиты шапки и табличные части, а также тип и длина номера документа. В Конфигураторе настраивается множество дополнительных параметров документов, например, возможность его проведения и перечень регистров, по которым он может делать движения, экранные формы и макеты (шаблоны печатных форм), последовательности, в которые входит данный документ и т.д.

Создание и проведение документа

В 1С:Предприятии 8.* можно программно создать, записать и провести документ. Для создания нового документа применяется метод СоздатьДокумент, который возвращает объект для записи типа ДокументОбъект.

Через другие объекты и механизмы нельзя создать, изменить или удалить документ.

Внимание! Программное проведение документа выполняется с помощью метода Записать с параметрами РежимЗаписиДокумента и РежимПроведенияДокумента.

Ниже приведены возможные значения параметров метода Записать: Параметр «РежимЗаписиДокумента»:

- Запись (по умолчанию),

- ОтменаПроведения,
- Проведение (в этом случае имеет смысл указывать второй параметр).

Параметр «РежимПроведенияДокумента»:

1. Неоперативный (по умолчанию),
2. Оперативный.

Поиск документа

Поиск документа по номеру производится методом НайтиПоНомеру. Этот метод возвращает ссылку на найденный документ или пустую ссылку, если документ не найден. Проверка на пустую ссылку производится методом Пустая().

Если требуется отобрать множество документов с определенным значением реквизита, то можно применить метод Выбрать с указанием структуры отбора и затем обойти полученную выборку.

Не забудьте, что у реквизита, по которому производится отбор, должен быть установлен в Конфигураторе флажок «Индексировать» (для даты документа можно считать, что он уже установлен).

Для выборки документов из базы данных по комбинированному условию, например, по нескольким полям, рекомендуется применять запросы (см. главу «Запросы»).

Удаление документа

Удаление документа производится с помощью объекта ДокументОбъект и только так. Другим способом удалить документ нельзя.

Внимание! При пометке на удаление методом УстановитьПометкуУдаления проведенный документ делается непроведенным и движения документа по регистрам удаляются (если не установлено свойство документа «Удалять движения автоматически»). А при присвоении свойству ПометкаУдаления значения Истина проведенный документ остается проведенным и движения документа по регистрам сохраняются.

При непосредственном удалении документа контроль за удалением движений документа возлагается на разработчика.

4.2 Задание

Выполнить лабораторную работу. Выполненная лабораторная работа оформляется в виде отчета (формат документа *.pdf) и прикрепляется к соответствующему элементу электронного курса.

4.2.1 Задание

В конфигурации создать документ ПриказНаУвольнение. Реквизит задать один – ФИО (Справочник.Сотрудники). Задать обработчик, который при проведении документа будет помещать сотрудника в группу «Уволенные» справочника Сотрудники. Учесть, что процедура пишется «&НаСервере», а обработка проведения вызывает основную функцию модуля.

4.3 Порядок выполнения работы

- 1) Создать в конфигураторе документ «ПринятиеНаРаботу» (рис.4.1).

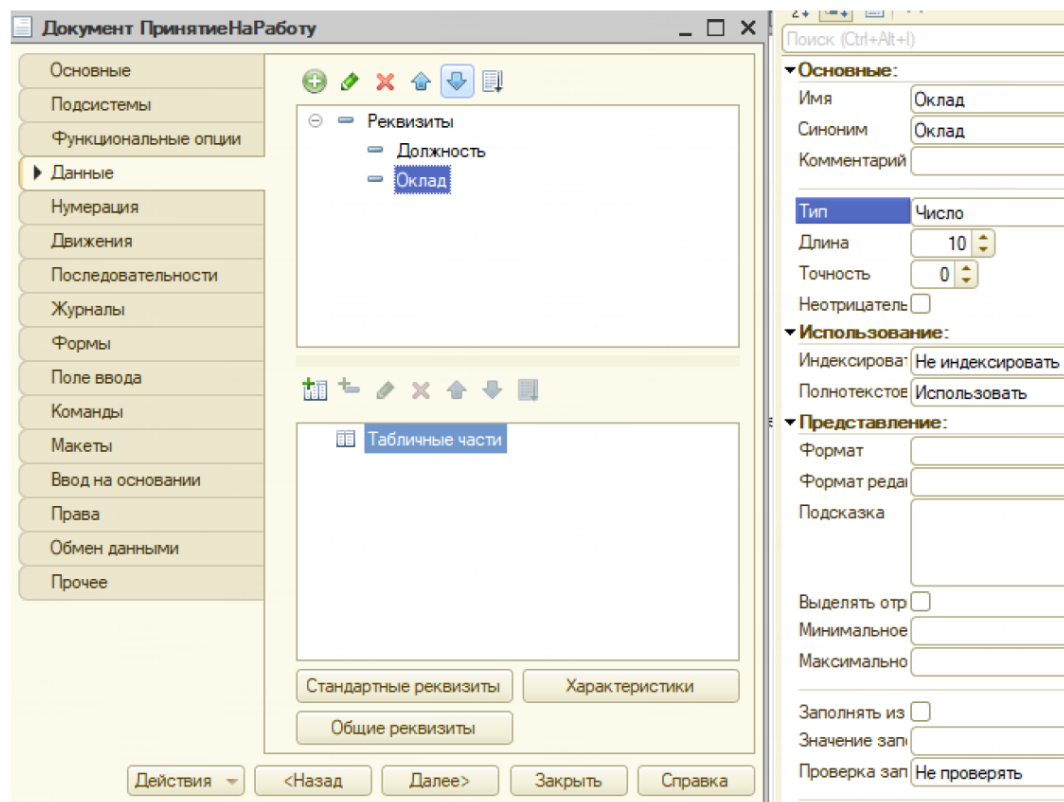


Рис.4.1 – Создание документа в 1С

Реквизиты документа:

- Должность (Справочник.Должности)
- Оклад (Число).

2) Нажать «Действия»- «Открыть модуль объекта» и задать обработчик.

```

Процедура ОбработкаПроведения()
Перем НовыйСотрудник;
    НовыйСотрудник = Новый Структура("Наименование, Должность, Оклад");
    НовыйСотрудник.Наименование = ФИО;
    НовыйСотрудник.Должность = Должность;
    НовыйСотрудник.Оклад = Оклад;
    Если СоздатьНовогоСотрудника(НовыйСотрудник) = 0 Тогда
        Сообщить("Не удалось создать нового сотрудника");
    КонецЕсли;
КонецПроцедуры

&НаСервере
Функция СоздатьНовогоСотрудника(СтруктураСправочника)
Перем НовыйСотрудник, СпрПользователи;
    СпрСотр = Справочники.Сотрудники;
    НовыйСотрудник = СпрСотр.СоздатьЭлемент();
    НовыйСотрудник.Наименование = СтруктураСправочника.Наименование;
    НовыйСотрудник.Должность = СтруктураСправочника.Должность;
    НовыйСотрудник.Оклад = СтруктураСправочника.Оклад;
    СсылкаНаГруппу =
Справочники.Сотрудники.НайтиПоНаименованию("Работающие");
    Если СсылкаНаГруппу.Пустая() Тогда
        НГ = Справочники.Сотрудники.СоздатьГруппу();
        НГ.Наименование = "Работающие";
        НГ.Записать();
        СсылкаНаГруппу = НГ.Ссылка;
    КонецЕсли;
    НовыйСотрудник.Родитель = СсылкаНаГруппу;
    Попытка
        НовыйСотрудник.Записать();
    Возврат 1;
Исключение
    Возврат 0;
КонецПопытки;
КонецФункции

```

4.4 Содержание отчета

1. Титульный лист, оформленный в соответствии с Образовательным стандартом ТУСУР (<https://regulations.tusur.ru/documents/70>).
2. На следующем листе сверху цель данной лабораторной работы.
3. Ход выполнения работы (содержащий пошаговое описание выполненных действий и скриншоты полученных результатов).

4. В отчете отразить скриншоты созданного документа, список элементов справочника Сотрудники до проведения документа и после, код обработчиков проведения.

5. Выводы.

Лабораторная работа №4 «Основы работы в 1С. Документы. Проведение»

Цель работы: знакомство с методами работы с документами в 1С и их проведением.

Темы для предварительного изучения: Печатные формы. Области. Макеты

5.1 Основы работы с печатными формами

Печатные формы

Для вывода информации, в таком виде, чтобы распечатать или сохранить в файл используют печатные формы. Для формирования печатных форм в 1С:Предприятии 8.0 предназначен табличный документ. С помощью элемента управления ПолеТабличногоДокумента он может быть отображен в экранной форме. Табличный документ также иногда используется для ввода данных в систему, поскольку его ячейки могут содержать значения, изменяемые с помощью элементов управления. Система 1С:Предприятие 8.0 имеет широкие возможности по формированию печатных форм. Обычно табличный документ формируется на основе макета, то есть шаблона. Построение макета производится в Конфигураторе в специальном табличном редакторе. Разработчик может создавать области, изменять высоту строк и ширину колонок, задавать форматирование ячеек и всего табличного документа, включать в макет рисунки, диаграммы и сводные таблицы.

На основе макета средствами встроенного языка формируется готовая печатная форма — табличный документ.

В большинстве случаев табличный документ предназначен для печати, и система предоставляет широкие возможности по настройке печати документа на

принтере можно задавать колонтитулы, масштабирование, поля от края листа, повторяющиеся на каждой странице («сквозные») строки и колонки, количество копий и другие опции.

Макеты

Макет представляет собой шаблон для формирования печатной формы. Макет тоже является табличным документом. Итоговая печатная форма строится на основе макета, как из строительных блоков. Для этого в макете создаются области, предназначенные для однократного или многократного вывода.

Сначала в табличный документ выводятся горизонтальные области Заголовок и Шапка. Ячейки табличного документа обычно содержат текст или параметры, а также могут содержать одновременно текст и параметры, заключенные в квадратные скобки. В последнем случае у таких ячеек свойство Заполнение должно быть установлено в значение «Шаблон». При формировании печатной формы средствами встроенного языка на место параметров будут подставлены конкретные значения.

Макеты могут быть у многих объектов конфигурации, например, у справочников, документов, отчетов, обработок и т.д. Кроме того, в конфигурации могут быть общие макеты, которые не принадлежат конкретному объекту, а располагаются в ветви «Общие макеты» дерева конфигурации.

Области

Областью табличного документа называется группа ячеек, имеющая имя. Обычно области включают в себя одну или несколько строк, колонок или любой прямоугольный диапазон ячеек. К области можно обратиться по координатам или имени. Именованная область Редактировать имена областей можно через окно свойств или с помощью панели «Имена», предварительно выделив нужный диапазон ячеек. Панель «Имена» имеет следующие кнопки:

- Отображение именованных строк / колонок,
- Отображение именованных ячеек,
- Назначить имя,
- Убрать имя.

Любая область уже имеет системное имя, например:

R5C3 — ячейка в 5-й строке и 3-й колонке;

R5 — 5-я строка;

C3 — 3-я колонка;

R5C4:R6C5 — диапазон ячеек;

C3:C5 — ячейки, входящие с 3-ю по 5-ю колонку;

R2:R6 — ячейки, входящие с 2-ю по 6-ю строки; T — табличный документ в целом;

D5 — 5-й рисунок (графический объект).

Разработчик может задать собственное имя для области, например, «Шапка», и использовать его при формировании табличного документа. Для вывода пересечения горизонтальной и вертикальной секции можно использовать конструкцию с вертикальной чертой, например, «Обл = Макет.ПолучитьОбласть("Строка | Колонка")». Второй вариант — задать имя для пересечения и работать с этой областью, как с любой другой.

Общий алгоритм связи макета с печатной областью выглядит так:

Алгоритм:

1. Создание макета в табличном редакторе (определение областей, имен параметров, форматирования).
2. Создание нового табличного документа (этот пункт не требуется, если отчет выводится в элемент управления ПолеТабличногоДокумента).
3. Получение макета в переменную (метод ПолучитьМакет).
4. Получение областей макета (метод ПолучитьОбласть).
5. Заполнение параметров области (свойство Параметры).
6. Вывод области в табличный документ (методы Вывести и Присоединить).
7. Установка свойств отображения табличного документа (сетка, заголовки, только просмотр и т.д.).
8. Отображение табличного документа на экране (а может быть отправка на принтер или сохранение).


```

    &НаКлиенте
    Процедура МатериалыКоличествоПриИзменении(Элемент)
        СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
        СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;
    КонiecПроцедуры

    &НаКлиенте
    Процедура МатериалыЦенаПриИзменении(Элемент)
        СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
        СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество * СтрокаТабличнойЧасти.Цена;
    КонiecПроцедуры

```

Рис.5.2. – Обработчик колонки «Сумма»

- 4) Создать Макет (печатную форму) для документа ПриходнаяНакладная (рис.5.3).

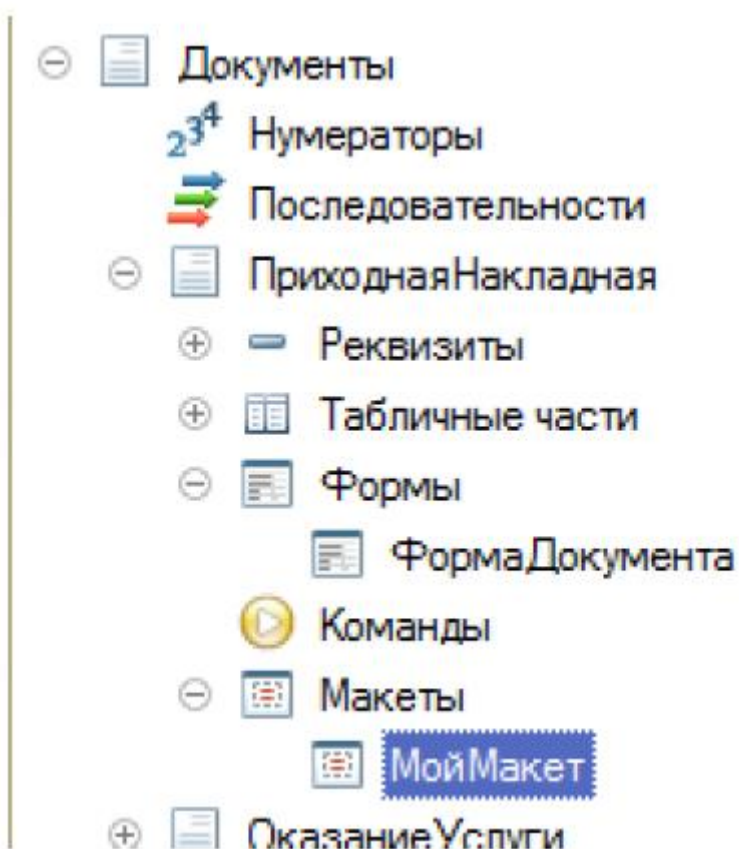


Рис.5.3 – Создание макета

- 5) Заполнить (рис.5.4) макет (<> - это Параметр)

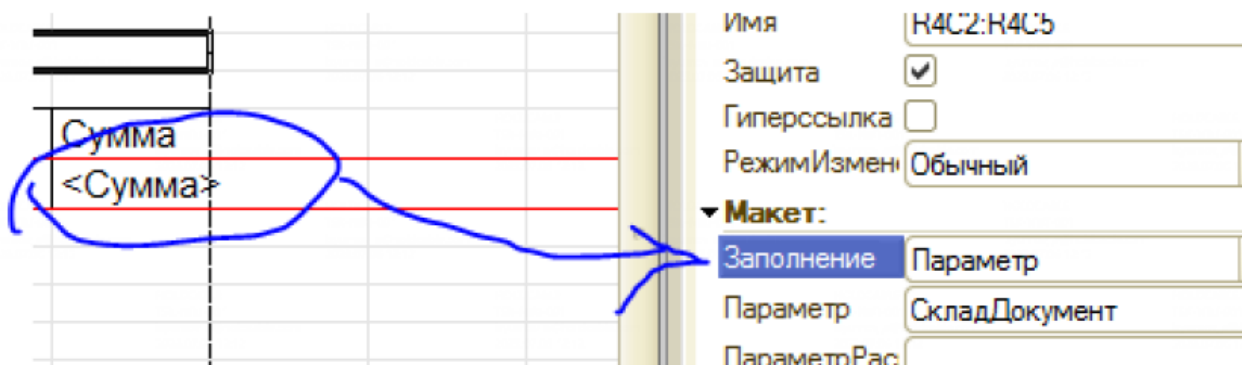


Рис.5.4 – Заполнение атрибута «Сумма»

6) Задать области через кнопку «Назначить имя» (рис.5.5).

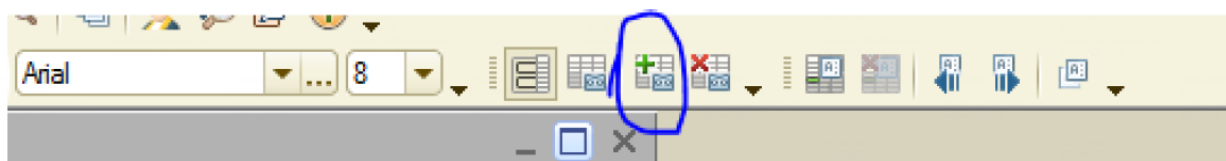


Рис.5.5. – Положение кнопки «Назначить имя» на панели инструментов

7) Для Макета задать 2 горизонтальные области: ОбластьШапка и ОбластьСтрока (рис.5.6).

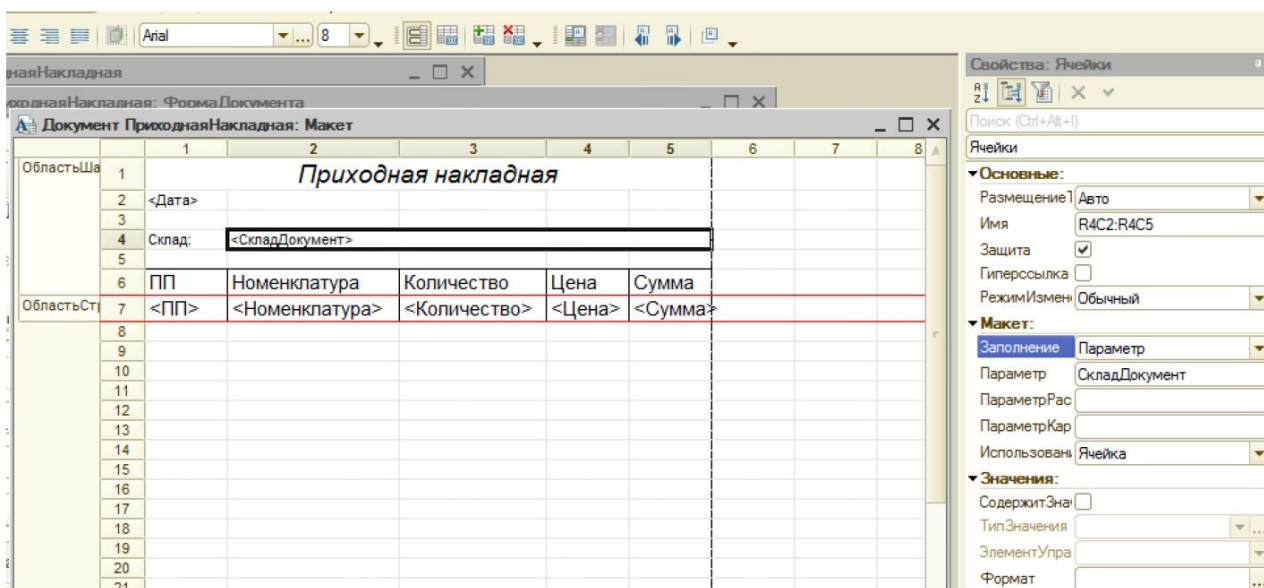


Рис.5.6. – Области ОбластьШапка и ОбластьСтрок

8) Задать обработчик кнопки «Распечатать» (рис.5.7).

```

□ Процедура Распечатать (Команда)
  // из алгоритма
  //2. Создание нового табличного документа
  ТД = Новый ТабличныйДокумент;
  ПодготовкаМакета (ТД);
  //8. Отображение табличного документа на экране
  ТД.Показать();
КонецПроцедуры
□ Процедура ПодготовкаМакета (ТД)
  //3. Получение макета в переменную
  Макет = Документы.ПриходнаяНакладная.ПолучитьМакет ("МойМакет");
  //4. Получение областей макета
  ОбластьШапка = Макет.ПолучитьОбласть ("ОбластьШапка");
  //5. Заполнение параметров области
  //Заполнить параметры в макете можно методом Заполнить, в данном случае
  //наименование параметров в макете и источнике (документе) должны совпадать
  ОбластьШапка.Параметры.Заполнить (Объект);
  //Если имена в макете и документе отличаются, то необходимо для каждого параметра
  //указать источник, из которого брать данные
  ОбластьШапка.Параметры.СкладДокумент = Объект.Склад;
  //6. Вывод области в табличный документ
  ТД.Вывести (ОбластьШапка);
  // при необходимости отрисовка других областей макет в печатной форма шаги 4 – 6 повторяем
  ОбластьДанных = Макет.ПолучитьОбласть ("ОбластьСтрока");
  Сч = 1;
  Для Каждого Строка из Объект.Материалы Цикл
    ОбластьДанных.Параметры.ПП = Сч; Сч = Сч+1;
    ОбластьДанных.Параметры.Номенклатура = Строка.Материал;
    ОбластьДанных.Параметры.Количество = Строка.Количество;
    ОбластьДанных.Параметры.Цена = Строка.Цена;
    ОбластьДанных.Параметры.Сумма = Строка.Сумма;
    ТД.Вывести (ОбластьДанных);
  КонецЦикла;
КонецПроцедуры

```

Рис.5.7. – Код обработчика кнопки

- 9) Для текущего макета добавить вертикальные области ОбластьПП, ОбластьНом, ОбластьЦифры (рис.5.8).

| | | ОбластьПП | ОбластьНом | ОбластьЦифры | | |
|-----------|---|----------------------------|-----------------|--------------|--------|---------|
| | | 1 | 2 | 3 | 4 | 5 |
| ОбластьШа | 1 | Приходная накладная | | | | |
| | 2 | <Дата> | | | | |
| | 3 | | | | | |
| | 4 | Склад: | <СкладДокумент> | | | |
| | 5 | | | | | |
| | 6 | ПП | Номенклатура | Количество | Цена | Сумма |
| ОбластьСт | 7 | <ПП> | <Номенклатура> | <Количество> | <Цена> | <Сумма> |
| | 8 | | | | | |
| | 9 | | | | | |

Рис.5.8 – Области ОбластьПП, ОбластьНом, ОбластьЦифры

- 10) Создать команду «РаспечататьПрисоединением» (рис.5.9).

```

        &НаКлиенте
    □ Процедура РаспечататьПрисоединением(Команда)
        // из алгоритма
        //2.    Создание нового табличного документа
        ТД = Новый ТабличныйДокумент;
        ПодготовкаМакетаПрисоединить(ТД);
        //8.    Отображение табличного документа на экране
        ТД.Показать();
    КонецПроцедуры

    □ Процедура ПодготовкаМакетаПрисоединить(ТД)
        //3.    Получение макета в переменную
        Макет = Документы.ПриходнаяНакладная.ПолучитьМакет("МойМакет");
        //4.    Получение областей макета
        ОбластьШапка = Макет.ПолучитьОбласть("ОбластьШапка");
        //5.    Заполнение параметров области
        //Заполнить параметры в макете можно методом Заполнить, в данном случае
        //наименование параметров в макете и источнике (документе) должны совпадать
        ОбластьШапка.Параметры.Заполнить(Объект);
        //Если имена в макете и документе отличаются, то необходимо для каждого параметра
        //указать источник, из которого брать данные
        ОбластьШапка.Параметры.СкладДокумент = Объект.Склад;
        //6.    Вывод области в табличный документ
        ТД.Вывести(ОбластьШапка);
        Сч = 1;
        Для Каждого Строка из Объект.Материалы Цикл
            ОбластьДанных = Макет.ПолучитьОбласть("ОбластьСтрока|ОбластьПП");
            ОбластьДанных.Параметры.ПП = Сч;    Сч = Сч+1;
            ТД.вывести(ОбластьДанных);
            //Вывести - отражает область со следующей строки
            ОбластьДанных = Макет.ПолучитьОбласть("ОбластьСтрока|ОбластьНом");
            ОбластьДанных.Параметры.Номенклатура = Строка.Материал;
            ТД.Присоединить(ОбластьДанных);
            //Присоединить - отражает область справа от уже отраженной области
            ОбластьДанных = Макет.ПолучитьОбласть("ОбластьСтрока|ОбластьЦифры");
            ОбластьДанных.Параметры.Количество = Строка.Количество;
            ОбластьДанных.Параметры.Цена = Строка.Цена;
            ОбластьДанных.Параметры.Сумма = Строка.Сумма;
            ТД.Присоединить(ОбластьДанных);
        КонецЦикла;
    КонецПроцедуры

```

Рис.5.9. – Обработчик команды «РаспечататьПрисоединением»

5.2.2 Задание №2

Создать отчет с формированием таблицы Пифагора в печатной форме..

5.2.3 Порядок выполнения работы

- 1) Сформировать печатную форму по предлагаемому на рис.5.10 образцу

← → ☆ Таблица пифагора

Кол колонок:

Кол строк:

Сформировать

Рис.5.10 – Форма отчета «Таблица Пифагора»

- 2) По заданным параметрам строк и колонок необходимо сформировать печатную форму с таблицей Пифагора (рис.5.11).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----|-------------------------|---|---|----|---|---|---|---|
| 1 | <i>Таблица Пифагора</i> | | | | | | | |
| 2 | | 1 | 2 | 3 | | | | |
| 3 | 1 | 1 | 2 | 3 | | | | |
| 4 | 2 | 2 | 4 | 6 | | | | |
| 5 | 3 | 3 | 6 | 9 | | | | |
| 6 | 4 | 4 | 8 | 12 | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |

Рис.5.11 – Печатная форма «Таблица Пифагора»

- 3) Обработчик кнопки «Сформировать» написать использованные в п.5.1.2 навыки.

5.4 Содержание отчета

1. Титульный лист, оформленный в соответствии с Образовательным стандартом ТУСУР (<https://regulations.tusur.ru/documents/70>).
2. На следующем листе сверху цель данной лабораторной работы.
3. Ход выполнения работы (содержащий пошаговое описание выполненных действий и скриншоты полученных результатов).

4. В отчете отразить скриншоты созданного документа, список элементов справочника Сотрудники до проведения документа и после, код обработчиков проведения. Привести скриншоты созданной печатной формы «Таблицы Пифагора», код обработчика кнопки «Сформировать» и вывод печатной формы для таблицы 7х5.
5. Выводы.

Список литературы

1. Образовательный стандарт вуза ОС ТУСУР 01-2021. Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления от 25.11.2021. - URL: <https://regulations.tusur.ru/documents/70> (дата обращения: 05.06.2023).
2. Краткое описание нотации BPMN. [Электронный ресурс] – URL: <https://habr.com/ru/companies/auriga/articles/667084> (дата обращения: 05.06.2023).
3. Изюмов А.А. Компьютерные технологии в науке и образовании / Изюмов, А.А., Коцубинский В.П. //Учебное пособие. - Томск: Эль Контент, 2012 - 150с. EDN: RBBRXP
4. Изюмов А. А. Компьютерные технологии в науке и образовании : учебное пособие / А. А. Изюмов, В. П. Коцубинский ; М-во образования и науки Российской Федерации, Томский гос. ун-т систем упр. и радиоэлектроники. - Томск : Эль Контент, 2012. - 148 с. : ил.; 29 см.; ISBN 978-5-4332-0024-1

Дополнительные источники для организации самостоятельного обучения

1. Аникеев С. В. Основы разработки приложений в среде 1С» (Аникеев, С. В. Основы разработки приложений в среде 1С : учебное пособие / С. В. Аникеев. — Рязань : РГРТУ, 2007. — 51 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/168352> (дата обращения: 06.07.2023).
2. Рыбалка В. В. Hello, 1С. Пример быстрой разработки приложений на 1С:Предприятие 8.3. Версия 3 / В. В. Рыбалка. — 3-е изд. — Москва : , 2014. — ISBN 978-5-9677-2101-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/251867> (дата обращения: 06.07.2023).
3. Шеер А. -. Индустрия 4.0: от прорывной бизнес-модели к автоматизации бизнес-процессов : учебник / А. -. Шеер. — Москва : Дело РАНХиГС, 2020. — 272 с. — ISBN 978-5-85006-194-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/171116> (дата обращения: 06.07.2023)
4. Зуева А. Н. Моделирование бизнес-процессов в нотации BPMN 2.0 : учебное пособие / А. Н. Зуева. — Москва : РТУ МИРЭА, 2021. — 105 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176564> (дата обращения: 06.07.2023).