РОСЖЕЛДОР

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»

(ФГБОУ ВО РГУПС)

И. А. Ольгейзер, О. В. Игнатьева, Е. В. Голубенко

ПРОГРАММИРОВАНИЕ В 1С

Учебно-методическое пособие
для лабораторных работ

Ростов-на-Дону
РГУПС
2022

УДК 004.4(07) + 06

Рецензент – кандидат технических наук, доцент В. В. Жуков

Ольгейзер, И. А.

Программирование в 1С : учебно-методическое пособие для лабораторных работ / И. А. Ольгейзер, О. В. Игнатьева, Е. В. Голубенко ; ФГБОУ ВО РГУПС. – Ростов-на-Дону : РГУПС, 2022. – 59 с.

Приведены задания для лабораторных работ по программированию в системе 1С:Предприятие. Рассматриваются задачи разработки и администрирования баз данных, автоматизации операций управления, учета и анализа с использованием платформы 1С и специального внедренного языка программирования.

Для студентов направлений «Информатика и вычислительная техника», «Информационные системы и технологии», изучающих дисциплины «Программирование в 1С», «Программное обеспечение отечественного производства», а также для всех обучающихся магистратуры, бакалавриата и специалитета различных направлений, изучающих смежные дисциплины и спецкурсы.

Одобрено к изданию кафедрой «Вычислительная техника и автоматизированные системы управления».

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1. СОЗДАНИЕ ПОДСИСТЕМ КОНФИГУРАЦИИ В УПРАВЛЯЕМОМ РЕЖИМЕ И ИНТЕРФЕЙСА В РЕЖИМЕ ОБЫЧНОГО ПРИЛОЖЕНИЯ

Цель лабораторной работы

Получить практические навыки в создании объекта конфигурации «Подсистемы» в режиме управляемого приложения и интерфейса в режиме обычного приложения.

Методические указания

Подсистемы – это основной инструмент построения командного интерфейса пользователя. Объекты метаданных «Подсистемы» имеют иерархическую структуру: чтобы настроить «подменю» в интерфейсе, необходимо добавить подчиненную подсистему. Отображение подсистем представлено на рис. 1.1.

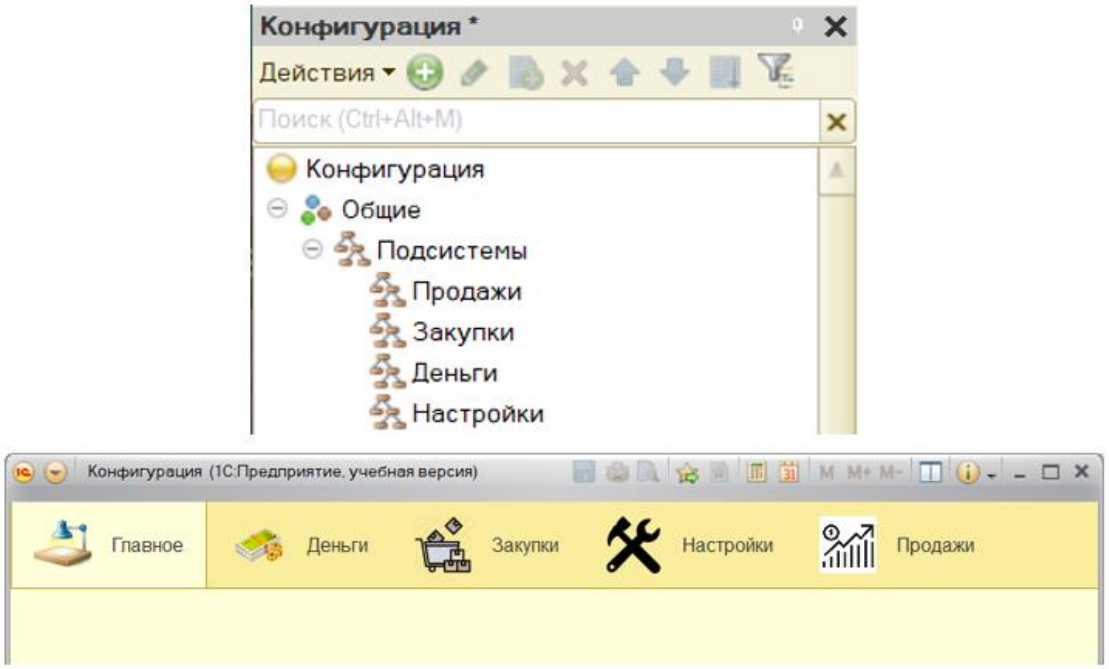

Рисунок 1.1 – Отображение подсистем в режиме конфигуратора и 1С:Предприятие

Для добавления подсистемы необходимо нажать правой клавишей мыши на вкладку «Подсистемы», выбрать пункты «Добавить» – «Подсистема». В этой же вкладке можно добавить подчиненную подсистему, чтобы создавать иерархические структуры (рис. 1.2).

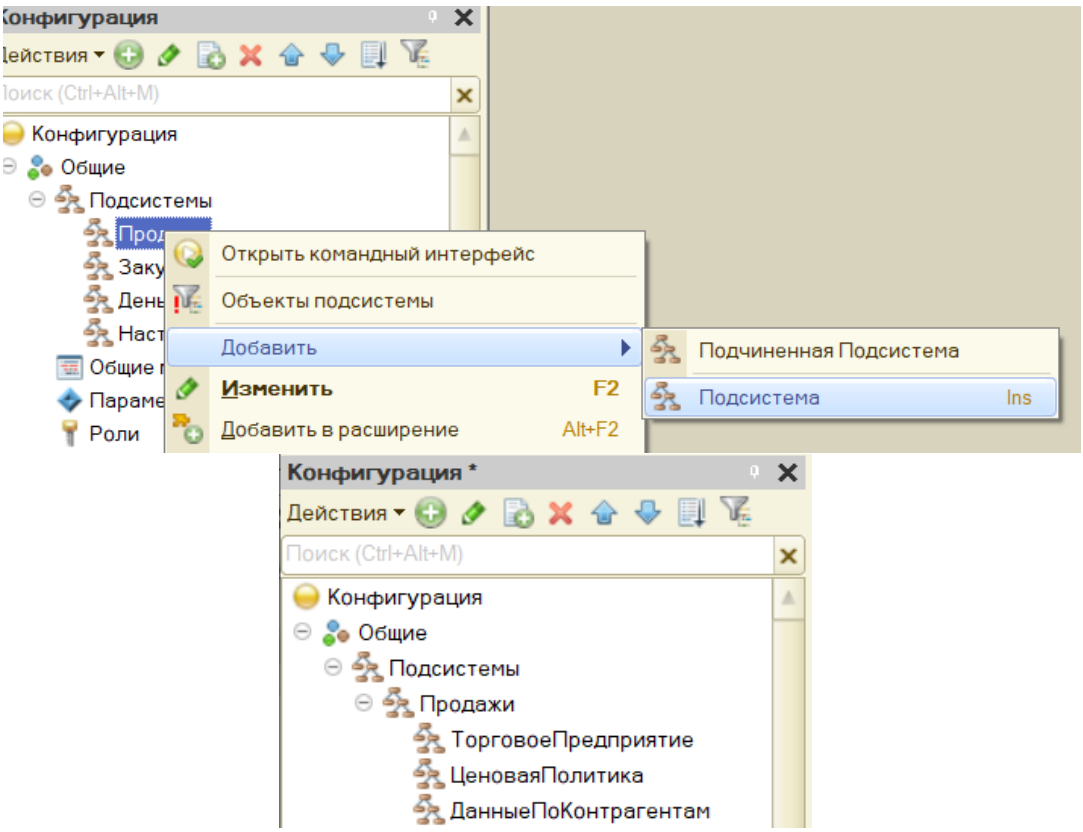

Рисунок 1.2 – Добавление подчиненной подсистемы

При создании подсистемы необходимо установить флаг «Включать в командный интерфейс», иначе подсистема не будет отображаться в интерфейсе (рис. 1.3).

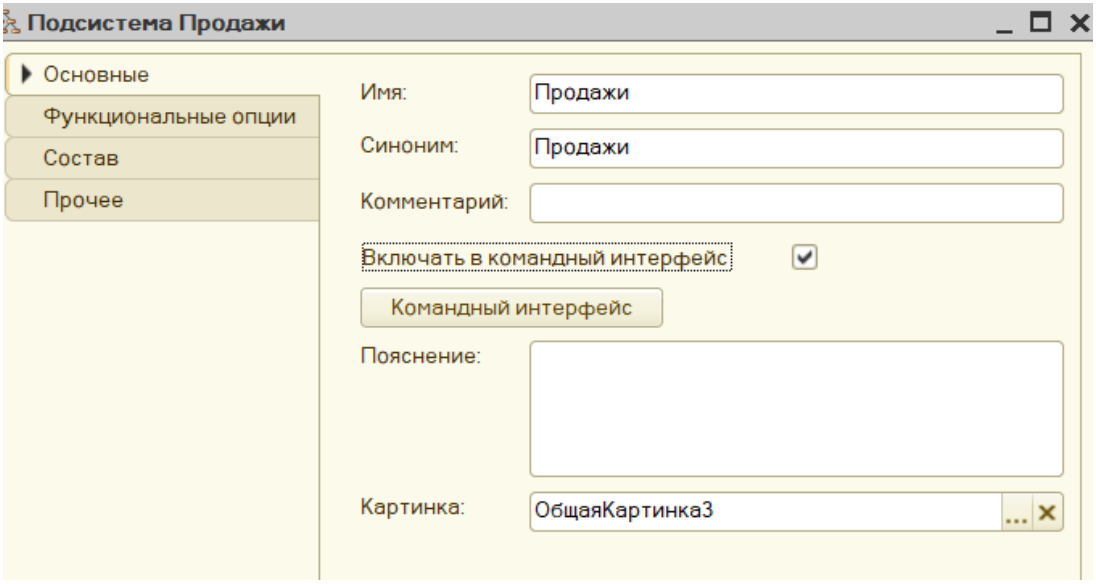

Рисунок 1.3 – Флаг «Включать в командный интерфейс»

Кнопка «Командный интерфейс» открывает панель настройки интерфейса, где можно настроить интерфейсы в зависимости от роли текущего пользователя (рис. 1.4).

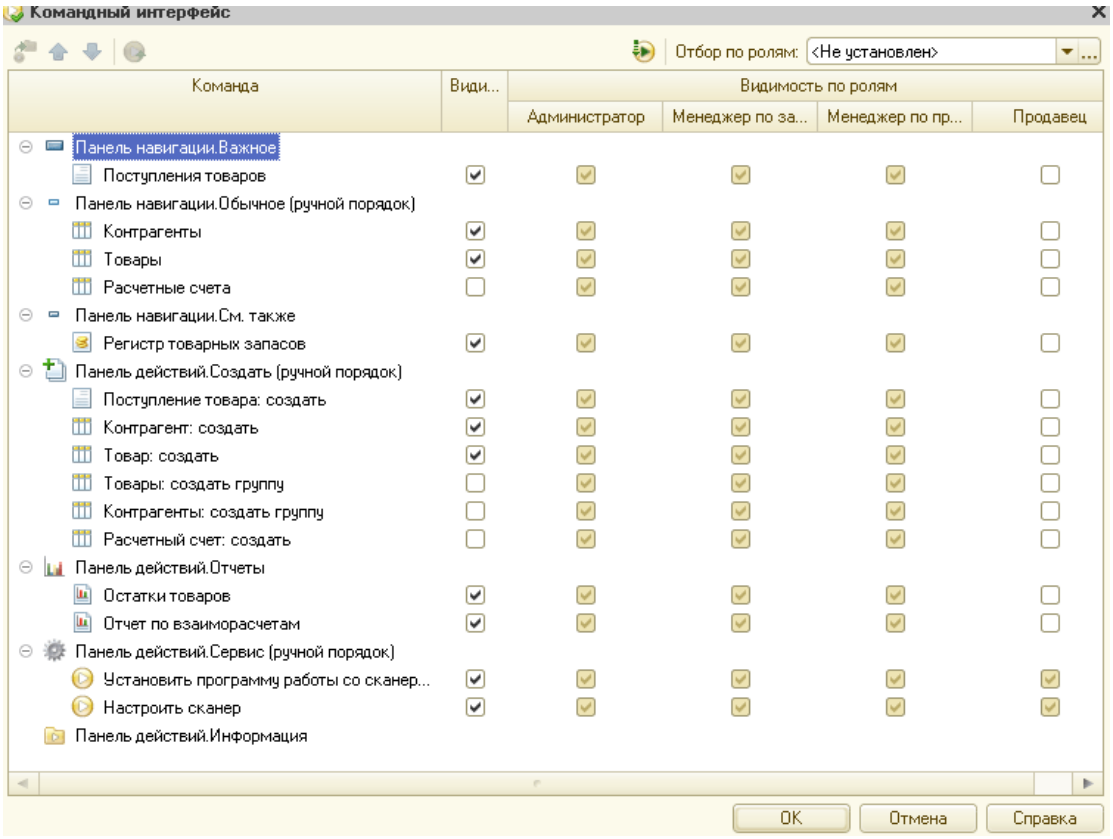

Рисунок 1.4 – Командный интерфейс

Создаваемые в конфигураторе объекты в зависимости от настройки могут отображаться в определенных подсистемах. Сделать это можно во вкладке «Свойства» (рис. 1.5).

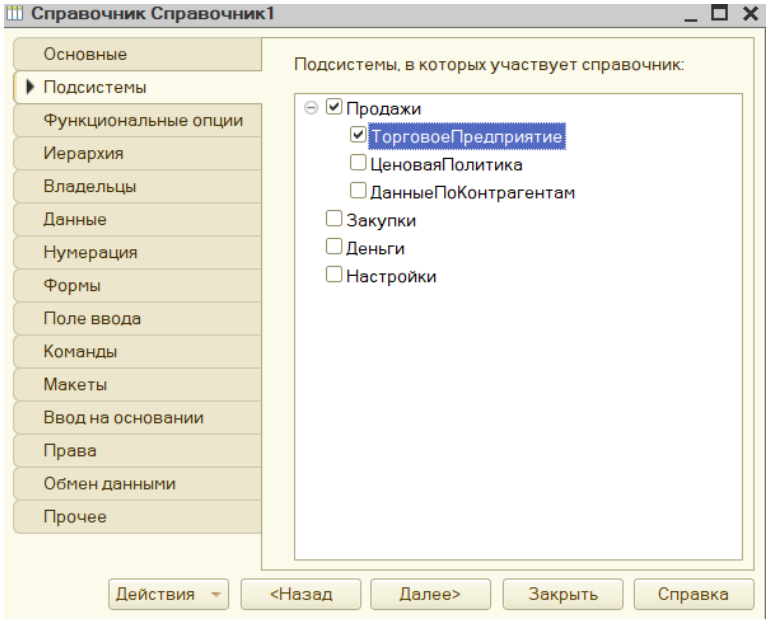

Рисунок 1.5 – Добавление объекта в подсистему

Картинка – изображение, назначенное для подсистемы, отображается в режиме «1С:Предприятие». Можно выбрать стандартную картинку, а можно добавить свою, предварительно создав её как объект конфигурации «Картинка» (рис. 1.6).

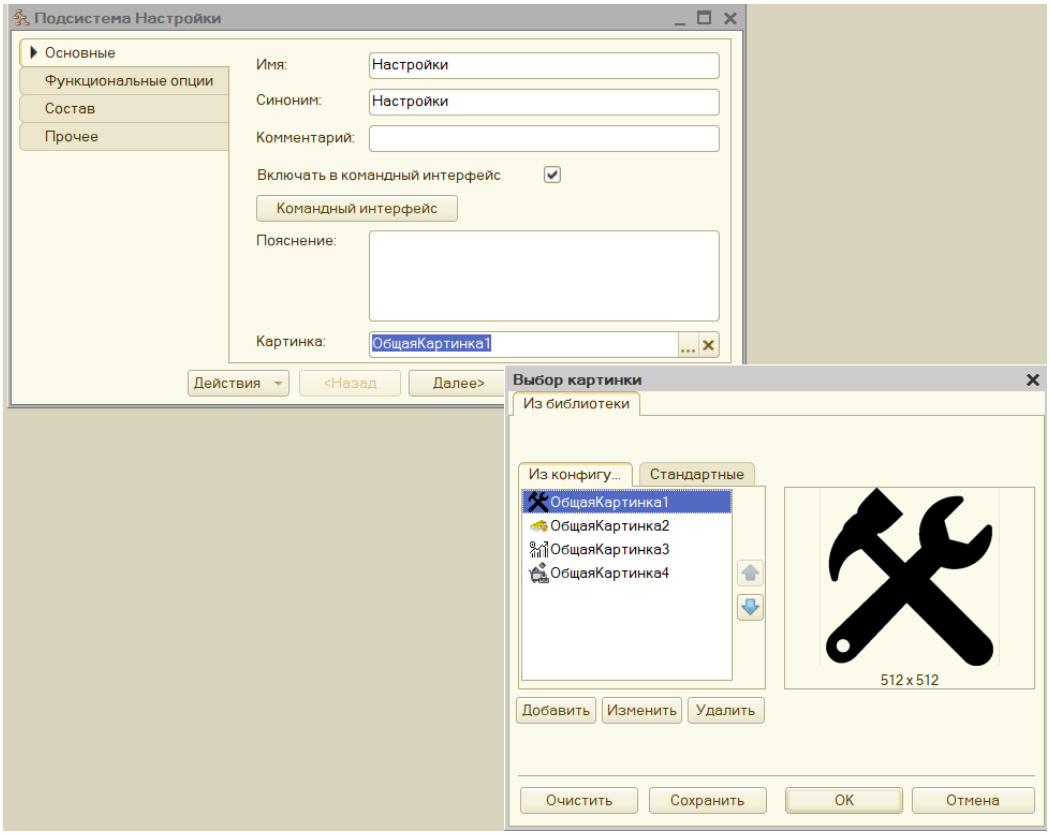

Рисунок 1.6 – Добавление картинки

На вкладке «Функциональные опции» указывается список функциональных опций, в которых используется данная подсистема.

Вкладка «Состав» определяет набор объектов метаданных, участвующих в данной подсистеме.

На вкладке «Прочее» можно описать справку к подсистеме и указать настройку «Включать в содержание справки» — включать ли данный раздел справки в общую справочную информацию по конфигурации.

Варианты заданий

1 Создать базу данных «Магазин бытовой техники», добавить в нее подсистемы «Товар», «Склады», «Продажи». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

2 Создать базу данных «Картинная галерея» и добавить в нее подсистемы «Картины», «Галереи», «Услуги». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

3 Создать базу данных «Швейная мастерская» и добавить в нее подсистемы «Товар», «Материал», «Услуги». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

4 Создать базу данных «Фитнес клуб» и добавить в нее подсистемы «Инвентарь», «Тренеры», «Секции», «Абонементы». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

5 Создать базу данных «Зоопарк» и добавить в нее подсистемы «Животные», «Вольеры», «Персонал». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

6 Создать базу данных «Колл-центр» и добавить в нее подсистемы «Сотрудники», «Отделы», «Офисы», «Заявки». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

7 Создать базу данных «Школа» и добавить в нее подсистемы «Сотрудники», «Ученики», «Предметы». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

8 Создать базу данных «Кинотеатр» и добавить в нее подсистемы «Афиши», «Сеансы», «Кинотеатры». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

9 Создать базу данных «Служба доставки» и добавить в нее подсистемы «Сотрудники», «Заказы», «Офисы», «Автомобили». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

10 Создать базу данных «Управление складами» и добавить в нее подсистемы «Склады», «Фирмы», «Товары», «Сотрудники». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

11 Создать базу данных «Военная база» и добавить в нее подсистемы «Базы», «Войска», «Сотрудники». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

12 Создать базу данных «Библиотека» и добавить в нее подсистемы «Сотрудники», «Посетители», «Литература». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

13 Создать базу данных «Университет» и добавить в нее подсистемы «Сотрудники», «Студенты», «Кафедры», «Дисциплины». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

14 Создать базу данных «Туристическая компания» и добавить в нее подсистемы «Офисы», «Сотрудники», «Клиенты», «Путевки». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

15 Создать базу данных «Аптека» и добавить в нее подсистемы «Препараты», «Сотрудники», «Продажи», «Рецепты». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

16 Создать базу данных «ГИБДД» и добавить в нее подсистемы «Сотрудники», «Отделы», «Штрафы». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

17 Создать базу данных «Аэропорт» и добавить в нее подсистемы «Сотрудники», «Техника», «Расписание», «Продажи». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

18 Создать базу данных «Ресторан» и добавить в нее подсистемы «Сотрудники», «Меню», «Товар», «Заказы». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

19 Создать базу данных «Сельское хозяйство» и добавить в нее подсистемы «Сотрудники», «Угодья», «Сельскохозяйственные культуры», «Журнал посевной». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

20 Создать базу данных «Обменный пункт» и добавить в нее подсистемы «Сотрудники», «Валюты», «Операции». Добавить картинки для отображения каждой из подсистем в режиме «1С:Предприятие».

ЛАБОРАТОРНАЯ РАБОТА № 2. ОПРЕДЕЛЕНИЕ РОЛЕЙ И АДМИНИСТРИРОВАНИЕ ПРАВ ПОЛЬЗОВАТЕЛЕЙ

Цель лабораторной работы

Получить практические навыки в создании ролей и администрировании прав пользователей.

Методические указания

Для управления доступа пользователей используется отдельный объект метаданных, который называется «Роли» (рис. 2.1).

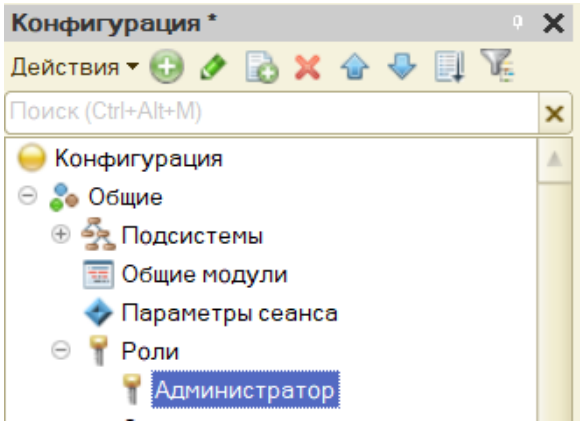

Рисунок 2.1 – Объект метаданных «Роли»

Роль определяет набор прав пользователя, которые он имеет. Для каждого из объектов (справочники, документы и т. д.) разработчик устанавливает свой набор прав: чтение, запись, добавление, изменение и пр.

Набор доступных прав – совокупность всех разрешений в ролях пользователя.

У объекта есть две закладки: «Права» и «Шаблоны ограничений». «Права» – основная закладка, «Шаблоны» – вкладка для настройки прав на уровне записи (рис. 2.2).

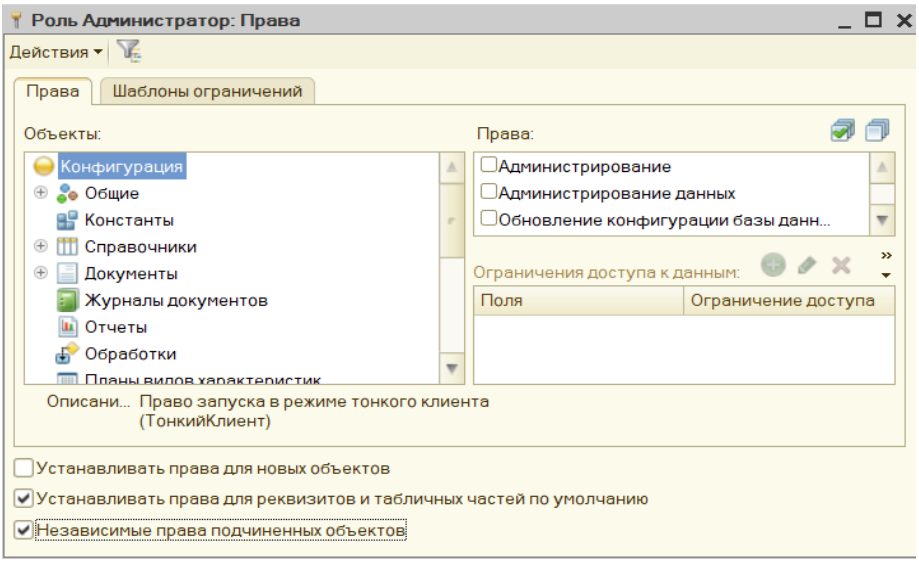

Рисунок 2.2 – Настройка роли

Объекты – это список метаданных, на которые будут устанавливаться права.

Права – список возможных для установки настроек прав.

Ограничение доступа к данным – поля роли для настройки РЛС (настроек прав на уровне записей).

Следует обратить внимание на флаги, расположенные в нижней части окна:

✓ Устанавливать права для новых объектов: если флаг установлен у роли, на новые объекты метаданных будут автоматически установлены разрешающие права;

✓ Устанавливать права для реквизитов и табличных частей по умолчанию: флаг, при установке которого реквизиты и табличные части будут наследовать права владельца (справочника, документа и т. д.);

✓ Независимые права подчиненных объектов: если флаг установлен, то система при определении права на объект конфигурации учтёт права на родительский объект.

Есть права в целом на конфигурацию, они устанавливаются, когда выделен корень дерева конфигурации в окне «Объекты конструктора роли». Точно так же для каждого объекта конфигурации определен свой набор прав. Его состав зависит от прототипа объекта (к примеру, для документов и справочников разный набор прав).

Для того чтобы назначить роль пользователю, необходимо в конфигураторе 1С открыть список пользователей: Главное меню – Администрирование – Пользователи (рис. 2.3).

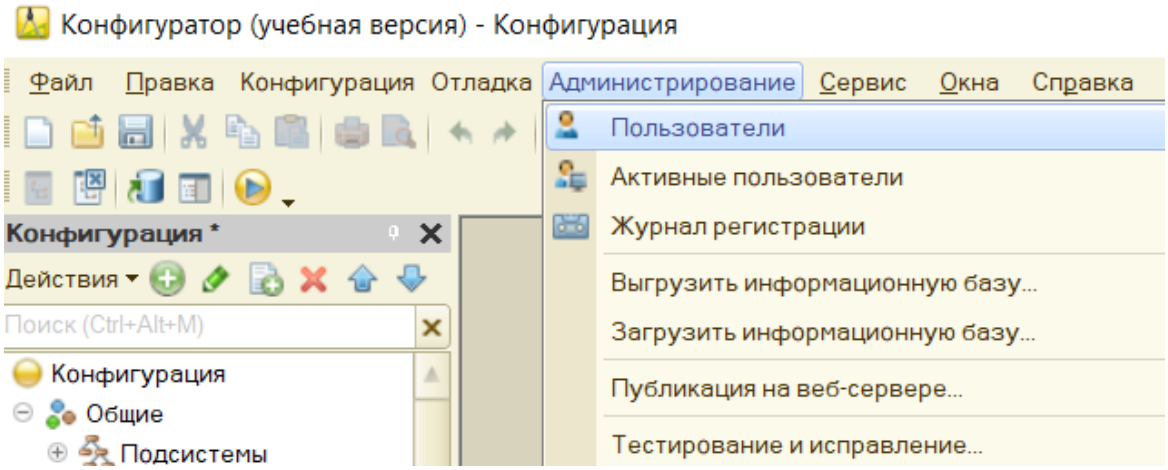

Рисунок 2.3 – Открытие списка пользователей

В открывшемся списке можно создать пользователя (рис. 2.4).

Далее необходимо задать Имя, Пароль и другие настройки (рис. 2.5).

Роли пользователю задаются на закладке «Прочие» (хотя бы один пользователь в базе данных должен быть с полными правами!)

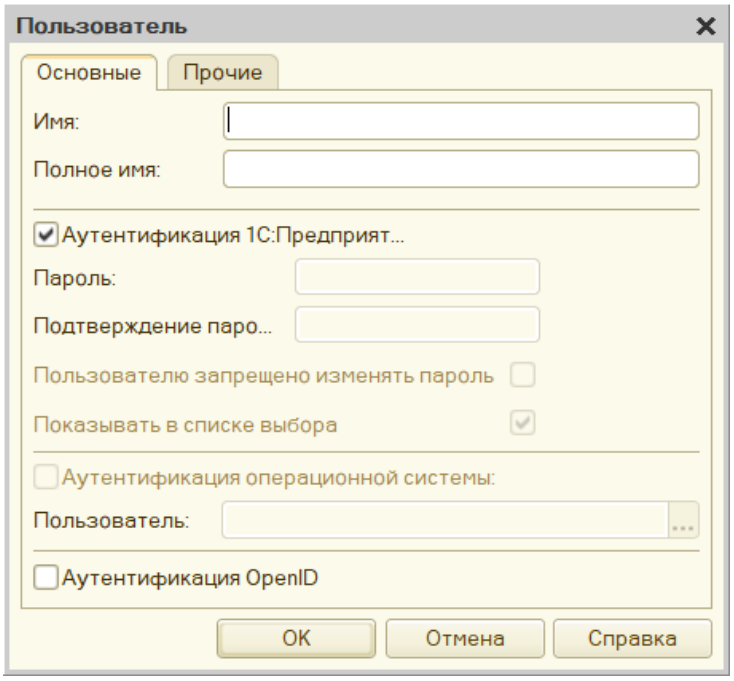

Рисунок 2.4 – Создание пользователя

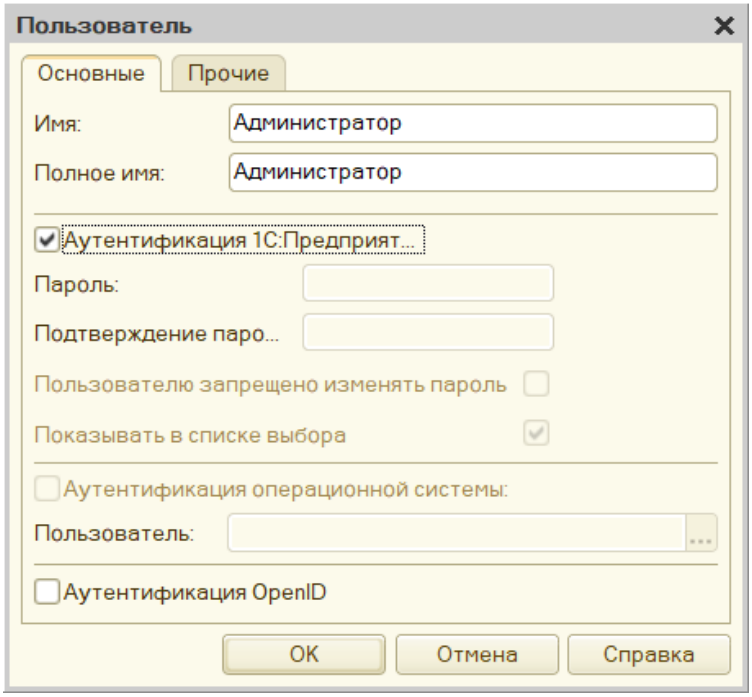

Рисунок 2.5 – Настройки нового пользователя

Варианты заданий

1 Создать в базе данных «Магазин бытовой техники» не менее трех ролей (администратор, директор, продавец), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

2 Создать в базе данных «Картинная галерея» не менее трех ролей (администратор, директор, галерист), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

3 Создать в базе данных «Швейная мастерская» не менее трех ролей (администратор, директор, швея), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

4 Создать в базе данных «Фитнес клуб» не менее трех ролей (администратор, директор, тренер), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

5 Создать в базе данных «Зоопарк» не менее трех ролей (администратор, директор, зоолог), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

6 Создать в базе данных «Колл-центр» не менее трех ролей (администратор, директор, оператор), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

7 Создать в базе данных «Школа» не менее трех ролей (администратор, директор, учитель), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

8 Создать в базе данных «Кинотеатр» не менее трех ролей (администратор, директор, кассир), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

9 Создать в базе данных «Служба доставки» не менее трех ролей (администратор, директор, доставщик), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

10 Создать в базе данных «Управление складами» не менее трех ролей (администратор, директор, кладовщик), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

11 Создать в базе данных «Военные базы» не менее трех ролей (администратор, начальник, подчиненный), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

12 Создать в базе данных «Библиотека» не менее трех ролей (администратор, директор, библиотекарь), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

13 Создать в базе данных «Университет» не менее трех ролей (администратор, декан, преподаватель), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

14 Создать в базе данных «Туристическая компания» не менее трех ролей (администратор, директор, турагент), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

15 Создать в базе данных «Аптека» не менее трех ролей (администратор, директор, фармацевт), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

16 Создать в базе данных «ГИБДД» не менее трех ролей (администратор, начальник, инспектор), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

17 Создать в базе данных «Аэропорт» не менее трех ролей (администратор, директор, диспетчер), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

18 Создать в базе данных «Ресторан» не менее трех ролей (администратор, директор, официант), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

19 Создать в базе данных «Сельское хозяйство» не менее трех ролей (администратор, директор, агроном), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

20 Создать в базе данных «Обменный пункт» не менее трех ролей (администратор, директор, кассир), определить для каждой из них права. Также создать для каждой роли не менее одного пользователя.

ЛАБОРАТОРНАЯ РАБОТА № 3. СОЗДАНИЕ ПРОСТЫХ И ИЕРАРХИЧЕСКИХ СПРАВОЧНИКОВ

Цель лабораторной работы

Получить практические навыки в создании простых и иерархических справочников.

Методические указания

Справочники в 1С используются для работы с постоянной или условно постоянной информацией, данная информация может содержать множество значений.

По умолчанию у любого справочника имеется два реквизита – это «Код» и «Наименование». Это так называемые стандартные реквизиты. Наименование обычно заполняет оператор, а код заполняется автоматически. Иногда стандартных реквизитов «Код» и «Наименование» бывает достаточно для работы, но чаще всего прикладная задача требует хранения вспомогательной информации. Для этого создаются реквизиты справочника, позволяющие хранить любую дополнительную информацию об элементе справочника.

Создание нового справочника. Для этого необходимо зайти в конфигуратор 1С, открыть окно конфигурации, выделить ветвь «Справочники», вызвать правой кнопкой мышки контекстное меню, где выполнить команду «Добавить» (рис. 3.1).

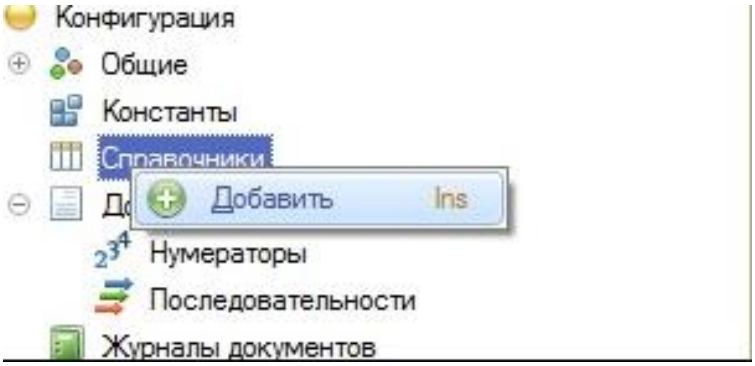

Рисунок 3.1 – Создание справочника

После этого откроется конструктор справочника, где на закладке «Основные» необходимо ввести имя справочника.

На закладке «Данные» можно задать длину кода и наименования, тип кода (число или строка) и основное представление элемента справочника (в виде кода или в виде строки). А также можно создать необходимые реквизиты и табличные части.

В качестве реквизитов справочника могут выступать, как примитивные типы, так и ссылочные (рис. 3.2).

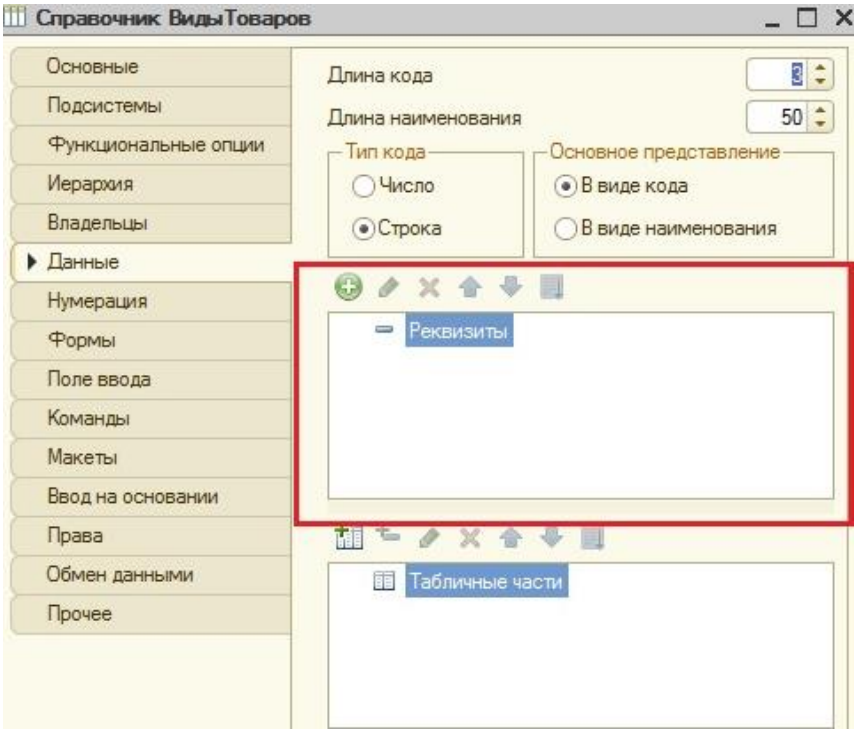

Рисунок 3.2 – Поле «Реквизиты»

В режиме «1С:Предприятие» добавление реквизита в справочник будет выглядеть следующим образом (рис. 3.3).

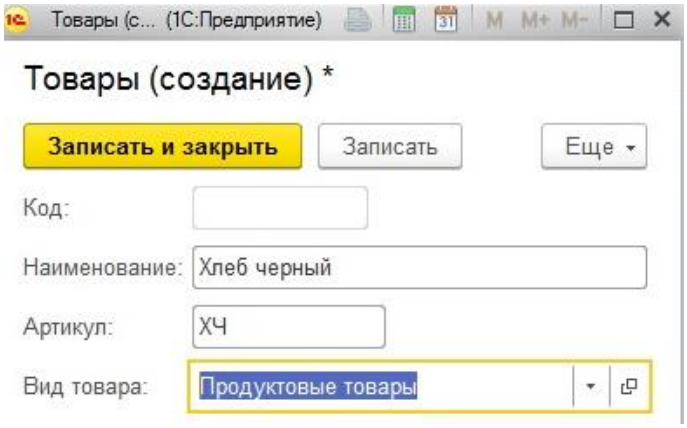

Рисунок 3.3 – Отображение реквизитов

Свойство «Основное представление» (закладка «Данные конструктора справочника»). Если основное представление в виде наименования, то представление элемента справочника в каком-либо реквизите будет в виде наименования этого элемента, как на рисунке представлен элемент справочника «Вид Товаров» в реквизите справочника «Товары». А если основное представление в виде кода, то тогда представление этого элемента будет в виде кода элемента.

Также можно создавать неограниченное количество табличных частей.

У табличной части может быть неограниченное количество реквизитов. Для того чтобы создать реквизит табличной части, её необходимо выделить и нажать на кнопку «Добавить реквизит» (рис. 3.4).

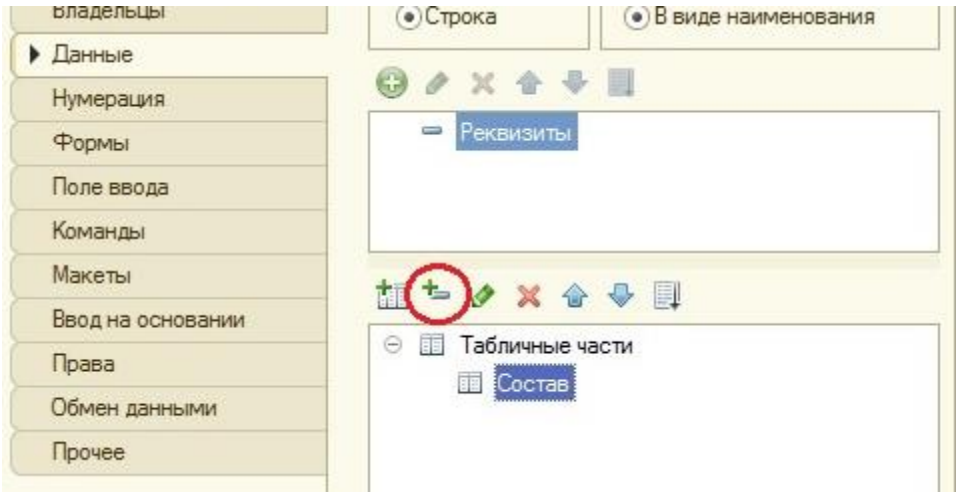

Рисунок 3.4 – Поле табличные части

В режиме «1С:Предприятие» добавление табличной части в справочник будет выглядеть следующим образом (рис. 3.5).

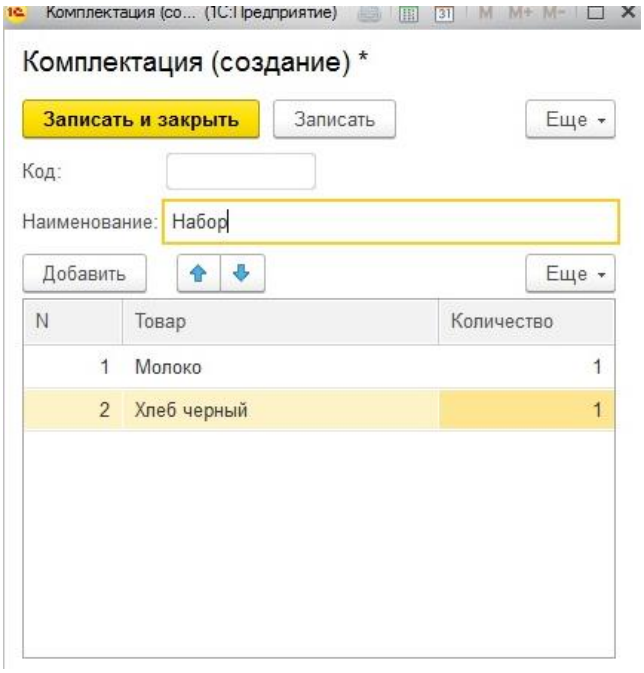

Рисунок 3.5 – Отображение табличных частей

В конфигурации 1С 8.3 можно разрабатывать подчиненные справочники. В подчиненном справочнике каждый его элемент имеет владельца, который является элементом или группой другого справочника. Элемент подчиненного справочника не может существовать без владельца.

Для создания подчиненного справочника нужно на вкладке «Владельцы» установить, кто является владельцем справочника (рис. 3.6).

Любой справочник можно сделать иерархическим – это значит, что пользователь сможет создавать каталоги (или, простым языком, папки), в которых будут содержаться элементы. Для этого необходимо перейти на закладку «Иерархия» конструктора справочника и установить флаг «Иерархический» (рис. 3.7)

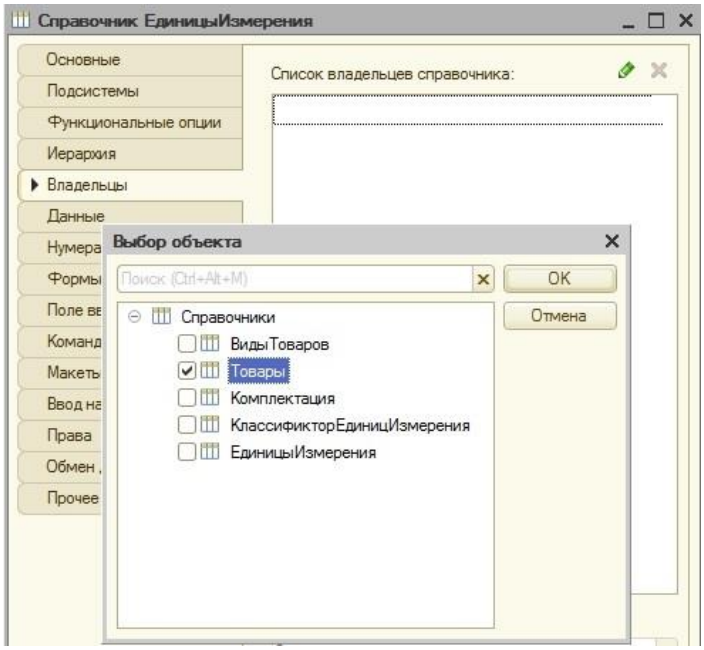

Рисунок 3.6 – Владельцы

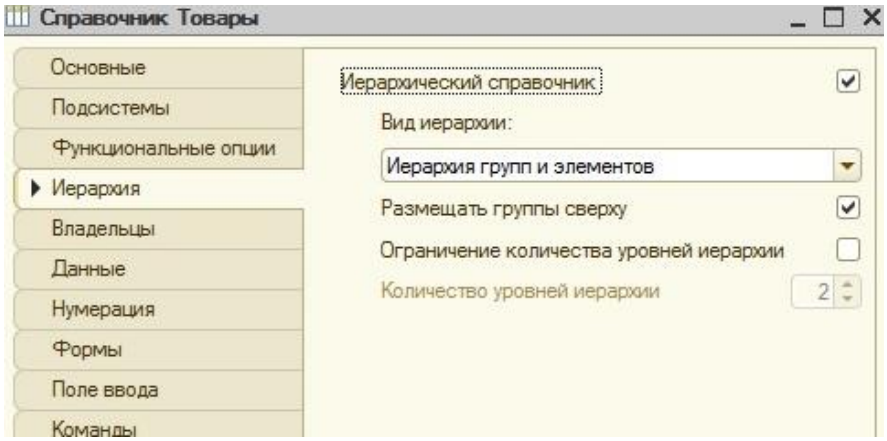

Рисунок 3.7 – Создание иерархии справочников

Иерархия у справочников бывает двух видов: «Иерархия групп и элементов» и просто «Иерархия элементов». «Иерархия групп» – это каталоги, которые в себе содержат определенные элементы. А «Иерархия элементов» – это когда один элемент подчинен другому.

После создания иерархии появится две команды: «Создать» и «Создать группу». При выполнении команды «Создать» будет открыта форма на создание нового элемента, а при выполнении команды «Создать группу» – форма на создание группы (папки) (рис. 3.8).

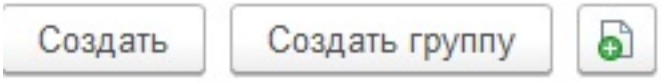

Рисунок 3.8 – Создание иерархии справочников

Варианты заданий

1 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Магазин бытовой техники»: «Перечень товара в магазине» (артикул, наименование, количество, цена, краткое описание, наличие на складе), «Перечень товара на складе» (артикул, наименование, количество, цена, краткое описание, номер склада), «Список складов» (номер склада, адрес), «Продажи» (количество, цена, итоговая стоимость, дата продажи). Добавить справочники в необходимые подсистемы.

2 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Картинная галерея»: «Перечень картин» (номер, название, год, автор, краткое описание, цена, выставлена/не выставлена), «Картины, переданные на другие выставки» (номер, название, год, автор, краткое описание, цена, номер галереи в которой находится), «Список галерей» (номер галереи, название, адрес), «Картины из других галерей» (номер, название, год, автор, краткое описание, цена, номер галереи, из которой привезена). Добавить справочники в необходимые подсистемы.

3 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Швейная мастерская»: «Перечень готового товара» (артикул, код материала, размеры, наименование, цена, количество), «Изделия на заказ» (наименование, размеры, код материала, краткое описание, цена, количество), «Материал» (код материала, название, плотность, стоимость), «Стоимость услуг» (код услуги, название услуги, стоимость, описание, время на исполнение). Добавить справочники в необходимые подсистемы.

4 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Фитнес-клуб»: «Перечень инвентаря» (артикул, наименование, цена, количество), «Список тренеров» (код тренера, ФИО, возраст, пол, дата принятия на работу, зарплата), «Секции» (название, время проведения, код тренера, стоимость), «Абонементы» (код абонемента, ФИО владельца, дата покупки, дата окончания, код тренера). Добавить справочники в необходимые подсистемы.

5 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Зоопарк»: «Перечень животных» (код вольера, наименование, количество), «Список вольеров» (код вольера, ФИО уборщика, тип вольера), «Типы вольеров» (код, наименование, особые условия, дополнительные сведения), «Персонал» (код сотрудника, ФИО, должность, стаж, дата принятия на работу). Добавить справочники в необходимые подсистемы.

6 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Колл-центр»: «Перечень сотрудников» (код, имя, фамилия, отчество, дата рождения, паспортные данные, адрес проживания, контактный номер, отдел), «Список отделов» (код,

наименование, офис, контактный номер, заведующий отделом), «Типы отделов» (код, наименование, особые условия, дополнительные сведения), «Список офисов» (код, город, наименование компании), «Заявки» (код, сотрудник, отдел, статус (заведена, отклонена, завершена и т. д.), дата, время изменения). Добавить справочники в необходимые подсистемы.

7 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Школа»: «Перечень сотрудников» (код, имя, фамилия, отчество, дата рождения, паспортные данные, адрес проживания, контактный номер, должность), «Перечень учеников» (код, имя, фамилия, отчество, дата рождения, адрес проживания, контактный номер родителей, класс), «Должность» (код, наименование, оклад), «Список предметов» (код, наименование, ведущий преподаватель). Добавить справочники в необходимые подсистемы.

8 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Кинотеатр»: «Афиша» (код, название фильма, режиссёр, дата премьеры, краткое описание), «Список сеансов» (номер сеанса, код кинотеатра, номер зала, фильм, цена билета, время сеанса), «Список кинотеатров» (код кинотеатра, адрес, телефон), «Список залов» (номер зала, количество мест). Добавить справочники в необходимые подсистемы.

9 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Служба доставки»: «Список доставщиков» (код доставщика, код автомобиля, ФИО, дата рождения, телефон, занят/свободен), «Заказы» (код заказа, код офиса, ФИО заказчика, код доставщика, цена заказа, дата заказа, комментарий), «Список офисов» (код офиса, адрес, телефон), «Список автомобилей» (код автомобиля, код офиса, номер, марка, год выпуска). Добавить справочники в необходимые подсистемы.

10 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Управление складами»: «Список складов» (номер склада, адрес склада, телефон, код фирмы, вместимость), «Фирмы» (код фирмы, название фирмы, адрес фирмы, телефон), «Товары» (код товара, код склада, наименование, количество, цена), «Работники» (ФИО, код склада, должность). Добавить справочники в необходимые подсистемы.

11 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Военные базы»: «Базы» (номер, код места расположение, вид войск, количество служащих), «Виды войск» (код, название, назначение), «Места расположения» (код места расположения, название, адрес, площадь), «Служащие» (ФИО, звание, дата рождения, номер базы, награды). Добавить справочники в необходимые подсистемы.

12 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Библиотека»: «Список сотрудников» (код, ФИО, должность, адрес, дата рождения, телефон), «Ви-

ды войск» (код, название, назначение), «Посетители» (код, ФИО, дата рождения, дата регистрации), «Список литературы» (код, жанр, автор, издательство, дата печати), «Жанры» (код, название). Добавить справочники в необходимые подсистемы.

13 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Университет»: «Студенты» (код, фамилия, имя, отчество, кафедра, год рождения, пол, адрес, город, телефон), «Преподаватели» (код, фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон), «Дисциплины» (код, название дисциплины, кафедра, кол-во часов, вид итогового контроля), «Кафедры» (код, название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон). Добавить справочники в необходимые подсистемы.

14 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Туристическая компания»: «Офисы» (номер офиса, наименование, адрес), «Туры» (код, название, страна, город, место проживания, цена), «Клиенты» (код, ФИО, дата рождения, пол, паспортные данные), «Проданные путевки» (код, номер офиса, ФИО клиента, название тура, стоимость, комментарий). Добавить справочники в необходимые подсистемы.

15 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Аптека»: «Препараты» (название, производитель, дозировка, вид, цена), «Сотрудники» (ФИО, дата рождения, дата поступления, образование), «Клиенты» (код, ФИО, дата рождения, пол, паспортные данные), «Продажи» (дата, лекарство, количество, номер рецепта, сотрудник), «Рецепты» (номер, дата выдачи, ФИО врача, ФИО покупателя). Добавить справочники в необходимые подсистемы.

16 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «ГИБДД»: «Сотрудники» (ФИО, Дата рождения, звание, дата поступления, выслуга лет, код отдела), «Отделы» (код отдела, название), «Клиенты» (код, ФИО, дата рождения, пол, паспортные данные), «Штрафы» (номер, вид правонарушения, ФИО нарушителя, ФИО сотрудника, размер штрафа), «Виды правонарушений» (код, название, статья). Добавить справочники в необходимые подсистемы.

17 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Аэропорт»: «Сотрудники» (фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон), «Самолеты» (номер, год выпуска, количество посадочных мест, грузоподъемность), «Расписание вылетов» (самолет, дата вылета, время вылета, место выбытия, место прибытия, стоимость билета), «Продажа билетов» (дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, количество билетов, стоимость). Добавить справочники в необходимые подсистемы.

18 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Ресторан»: «Сотрудники» (ФИО, должность, дата рождения, дата принятия, зарплата), «Меню» (название, описание, цена), «Продукты» (код, название, количество, поставщик, цена, срок годности), «Журнал заказов» (номер заказа, блюдо, ФИО официанта, номер столика, стоимость). Добавить справочники в необходимые подсистемы.

19 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Сельское хозяйство»: «Сотрудники» (ФИО, должность, дата рождения, дата принятия, зарплата), «Список культур» (название, вид, время высаживания, время созревания), «Список угодий» (номер, адрес, площадь, вид почвы, владелец), «Журнал посевной» (дата посева, номер участка, название растения, период полива, удобрения, особые условия). Добавить справочники в необходимые подсистемы.

20 Создать следующие справочники (обязательно использование подчиненных и иерархических справочников) в базе данных «Обменный пункт»: «Сотрудники» (код, ФИО, должность, дата рождения, дата принятия, зарплата), «Виды валют» (код, название, страна), «Курсы валют» (вид валюты, новая цена, старая цена, разница), «Операции обмена» (номер операции, ФИО сотрудника, ФИО клиента, дата операции, сумма операции). Добавить справочники в необходимые подсистемы.

ЛАБОРАТОРНАЯ РАБОТА № 4. ДОБАВЛЕНИЕ ДОПОЛНИТЕЛЬНЫХ РЕКВИЗИТОВ, ССЫЛОЧНЫЕ РЕКВИЗИТЫ

Цель лабораторной работы

Получить практические навыки в добавлении дополнительных реквизитов и создании ссылочных реквизитов.

Методические указания

Механизм «Дополнительные реквизиты» был разработан для того, чтобы любой пользователь, не прибегая к помощи программистов, мог самостоятельно добавить какую-то дополнительную информацию в элементы справочников.

Для добавления необходимо открыть элемент нужного справочника, найти на форме кнопку «Еще» и в выпадающем списке выбрать пункт «Изменить состав дополнительных реквизитов» (рис. 4.1).

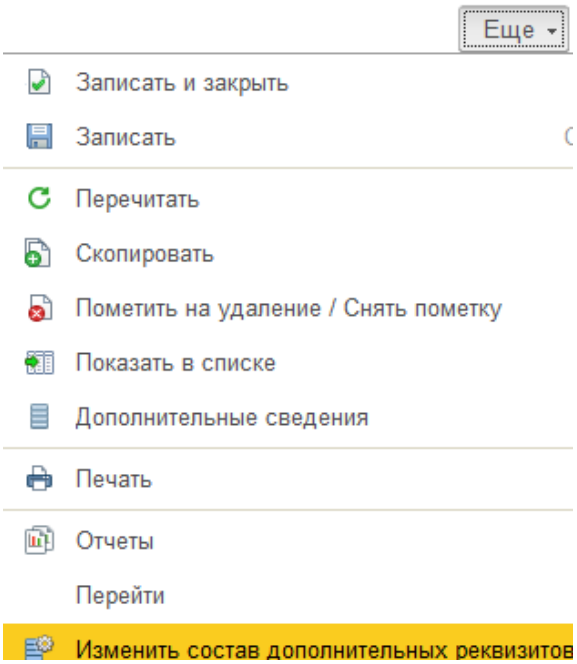

Рисунок 4.1 – Добавление нового реквизита

Далее откроется форма, в которой можно увидеть все дополнительные реквизиты, которые уже есть в программе. В левой части окна справочники находятся владельцы дополнительных реквизитов, в правой части – сами дополнительные реквизиты (рис. 4.2).

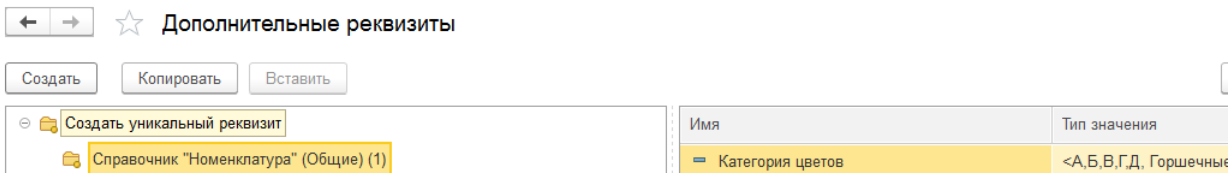

Рисунок 4.2 – Окно дополнительных реквизитов

Далее необходимо нажать кнопку «Создать». Откроется форма, в которой нужно задать все параметры нового реквизита, а именно:

✓ наименование (отражает суть нового реквизита);

✓ тип значения (какого вида значение будет содержаться в этом дополнительном реквизите, число, строка, значение какого-то другого справочника или же одно из заданных заранее значений);

✓ если выбран тип «Дополнительное значение», то дальше нужно перейти на вкладку «Значения» и нажимая кнопку «Создать», внести все необходимые варианты значения реквизита.

Чтобы сохранить, необходимо нажать «Записать и закрыть» (рис. 4.3).

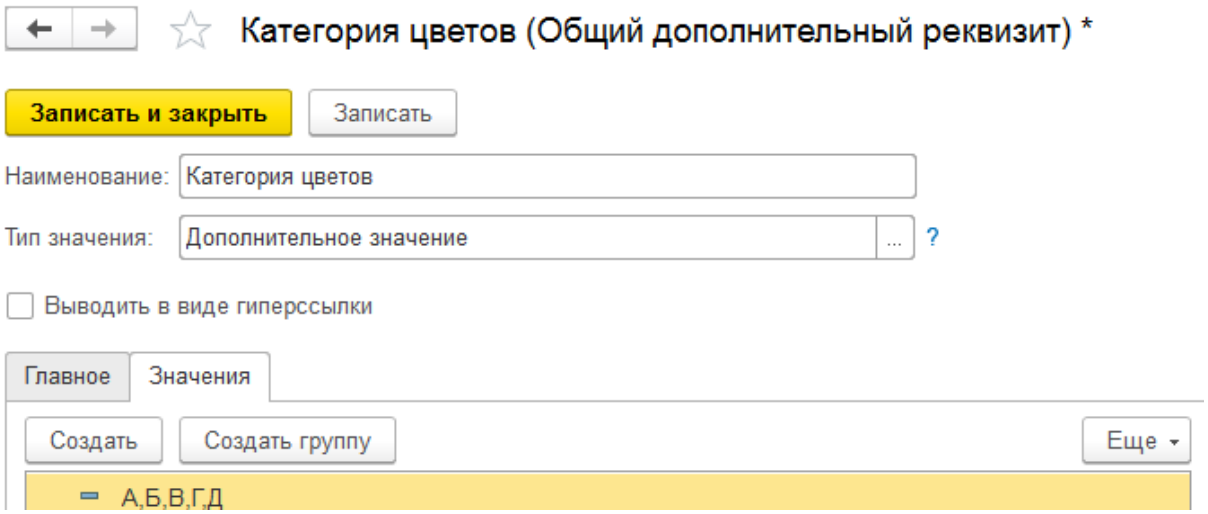

Рисунок 4.3 – Создание дополнительных реквизитов

После проделанных действий при открытии элементов справочника будут отображаться созданные дополнительные реквизиты.

Варианты заданий

Задание для всех вариантов: добавить в созданные в предыдущей лабораторной работе справочники дополнительные реквизиты.

ЛАБОРАТОРНАЯ РАБОТА № 5. НАПИСАНИЕ ПРОСТЫХ ЗАПРОСОВ И ПОЛЬЗОВАТЕЛЬСКАЯ НАСТРОЙКА ОТЧЕТОВ

Цель лабораторной работы

Получение практических навыков в написании простых запросов и пользовательской настройки отчетов.

Методические указания

Отчеты

Отчеты – это прикладные объекты конфигурации. Они предназначены для обработки накопленной информации и получения сводных данных в удобном для просмотра и анализа виде. Конфигуратор позволяет формировать набор различных отчетов, достаточных для удовлетворения потребности пользователей системы в достоверной и подробной выходной информации.

Как правило, для формирования выходных данных отчет использует систему компоновки данных. Но, вообще говоря, отчет может содержать произвольный алгоритм формирования «бумажного» или «электронного» отчета на встроенном языке.

Механизм запросов

Механизм запросов – это один из способов доступа к данным, которые поддерживает платформа. Используя этот механизм, разработчик может читать и обрабатывать данные, хранящиеся в информационной базе; изменение данных с помощью запросов невозможно. Это объясняется тем, что запросы специально предназначены для быстрого получения и обработки некоторой выборки из больших массивов данных, которые могут храниться в базе данных.

Для того чтобы разработчик имел возможность использовать запросы для реализации собственных алгоритмов, в платформе реализован язык запросов. Этот язык основан на SQL, но при этом содержит значительное количество расширений, ориентированных на отражение специфики финансово-экономических задач и на максимальное сокращение усилий по разработке прикладных решений.

Используя меню «Файл» и вкладку «Новый», можно добавить новый внешний отчет, дать ему имя и сохранить на диск. Далее нужно создать схему компоновки, используя кнопку «Открыть схему компоновки данных» (рис. 5.1).

После создания схемы компоновки нужно написать запрос, который будет собирать данные для отчета. Для этого на закладке «Наборы данных» нужно «Добавить набор данных – запрос» (рис. 5.2).

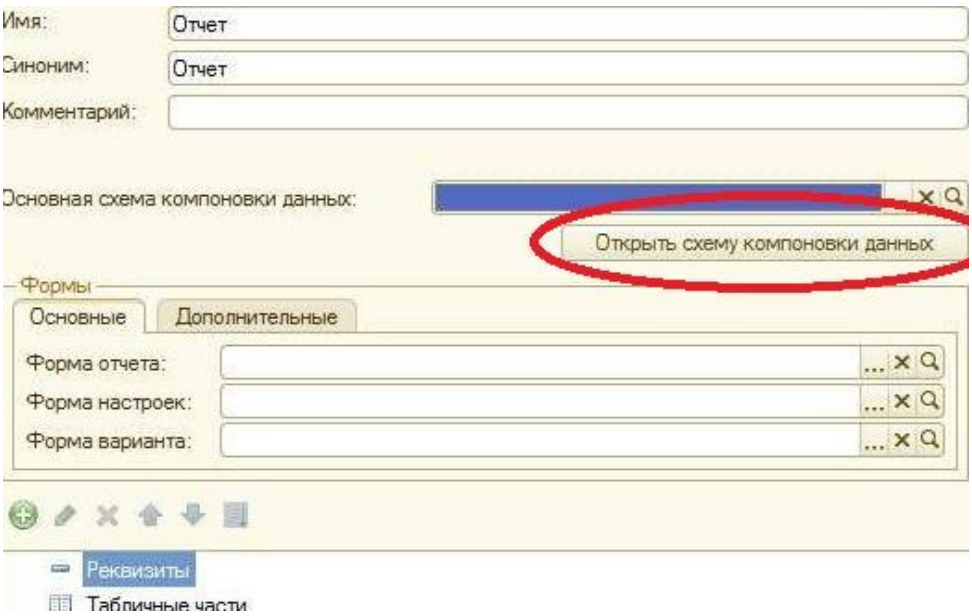

Рисунок 5.1 – Открытие схемы компоновки данных

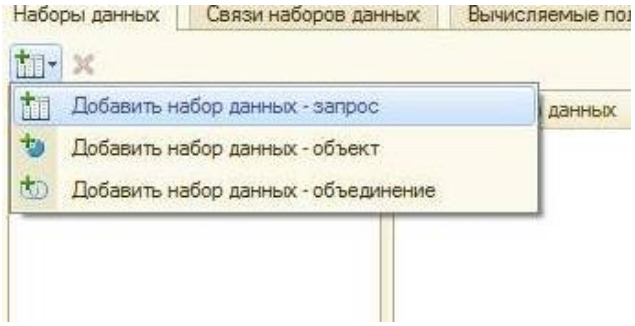

Рисунок 5.2 – Создание запроса

Если поля какой-либо таблицы имеют ссылочный тип (хранят ссылки на объекты другой таблицы), разработчик может в тексте запроса ссылаться на них через «.», при этом количество уровней вложенности таких ссылок система не ограничивает (рис. 5.3).

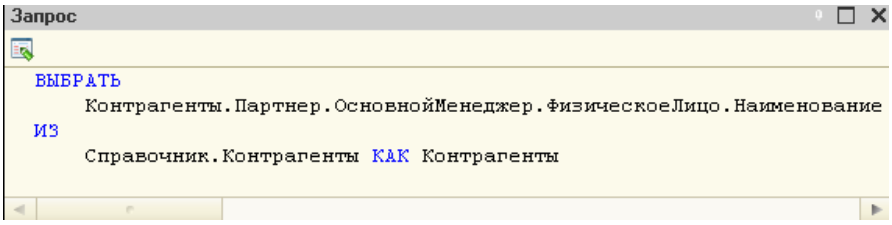

Рисунок 5.3 – Пример простого запроса

Система поддерживает обращения к вложенным табличным частям и как к отдельным таблицам, и как к целым полям одной таблицы. Например, при обращении к документу «Реализация товаров» (содержащему табличную часть «Товары» с составом отгружаемых товаров), мы можем считать табличную часть как отдельную таблицу (рис. 5.4).

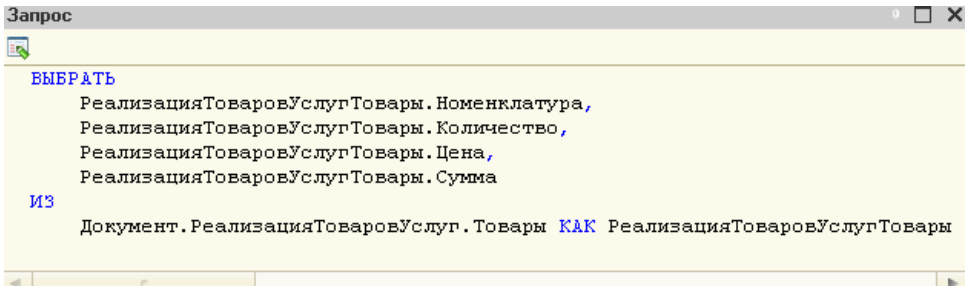

Рисунок 5.4 – Пример простого запроса с табличной частью

После создания отчета перейдем к пользовательской настройке.

Окно «продвинутых» настроек варианта отчета вызывается по команде «Еще» - «Прочее» - «Изменить вариант отчета» (рис. 5.5).

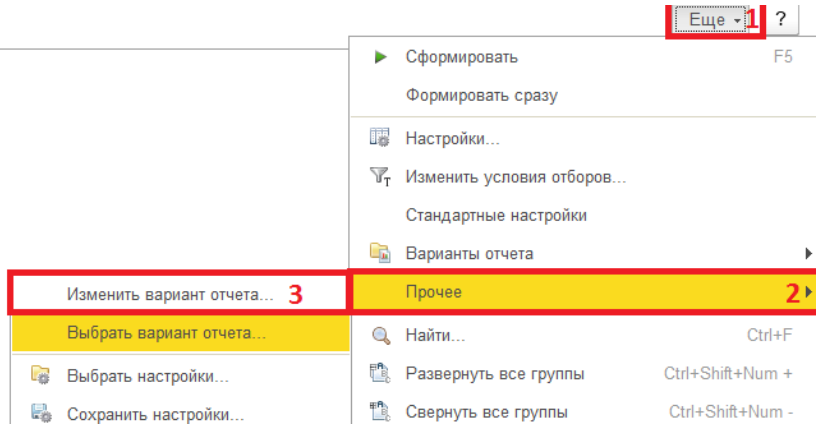

Рисунок 5.5 – Вызов «продвинутых» настроек

Окно изменения варианта отчета разделено на две части: Структура отчета, Настройки отчета (рис. 5.6).

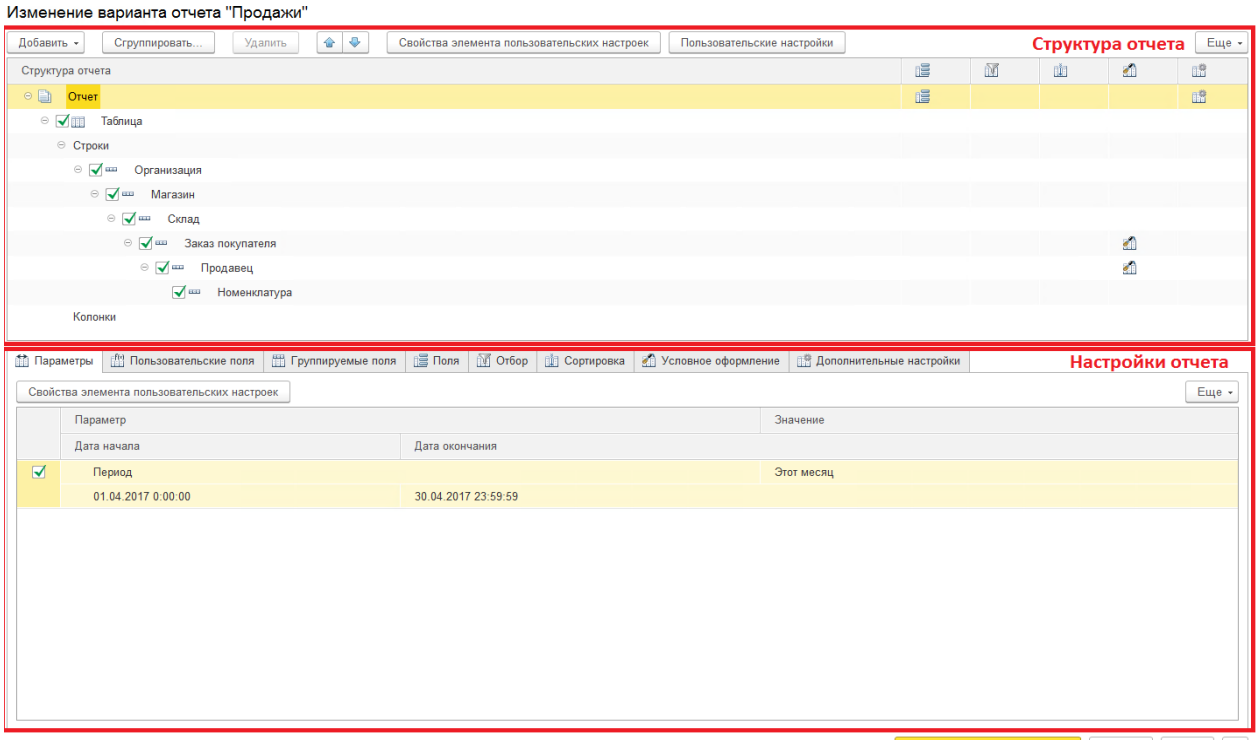

Рисунок 5.6 – Окно пользовательских настроек

Таблица структуры варианта отчета, кроме непосредственно колонки с группировками, содержит несколько дополнительных колонок:

✓ «Наличие выбранных полей в элементе структуры» (значок в данной колонке указывает на то, что в группировке используется особая настройка выводимых полей);

✓ «Наличие отбора в элементе структуры» (значок в данной колонке указывает на то, что в группировке используется особая настройка отборов);

✓ «Наличие сортировки в элементе структуры» (значок в данной колонке указывает на то, что в группировке используется особая настройка сортировки);

✓ «Наличие условного оформления в элементе структуры» (значок в данной колонке указывает на то, что в группировке используется особая настройка условного оформления);

✓ «Наличие других настроек в элементе структуры» (значок в данной колонке указывает на то, что в группировке используется особая конфигурация дополнительных настроек).

Раздел настроек состоит из следующих вкладок:

– Параметры. Содержит параметры СКД, доступные пользователю (рис. 5.7).

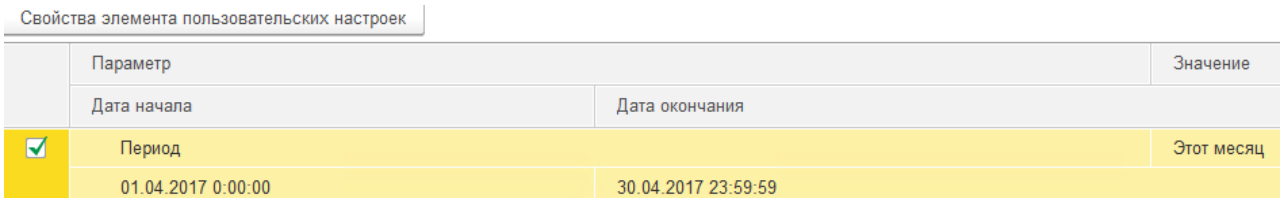

Рисунок 5.7 – Раздел «Параметры»

– Пользовательские поля. Содержит поля, которые формирует сам пользователь на основании данных, выбираемых отчетом (рис. 5.8).

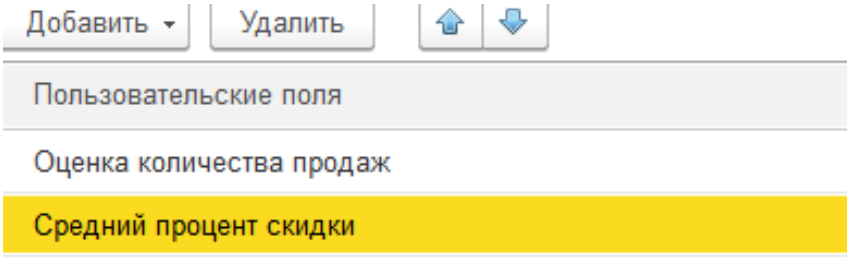

Рисунок 5.8 – Раздел «Пользовательские поля»

– Группируемые поля. Содержит поля, по которым будет группироваться результат варианта отчета (рис. 5.9).

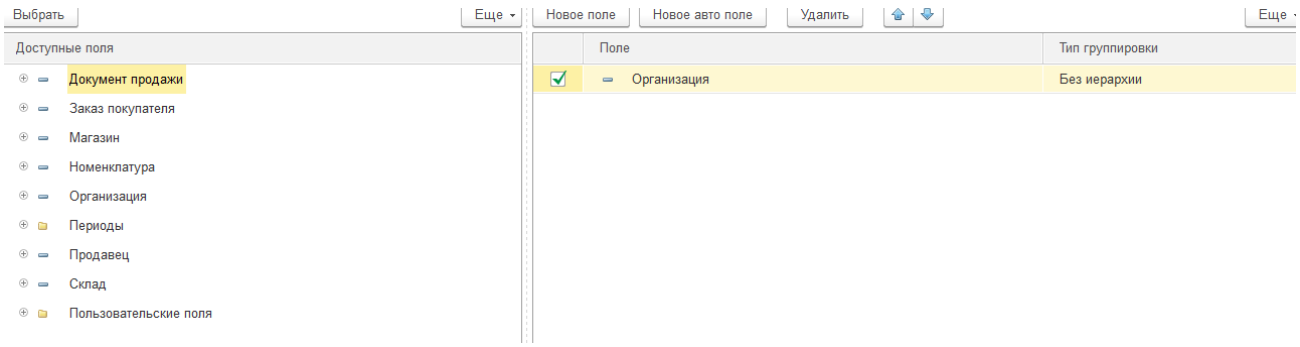

Рисунок 5.9 – Раздел «Группируемые поля»

– Поля. Содержит поля, которые будут выведены в результат варианта отчета (рис. 5.10).

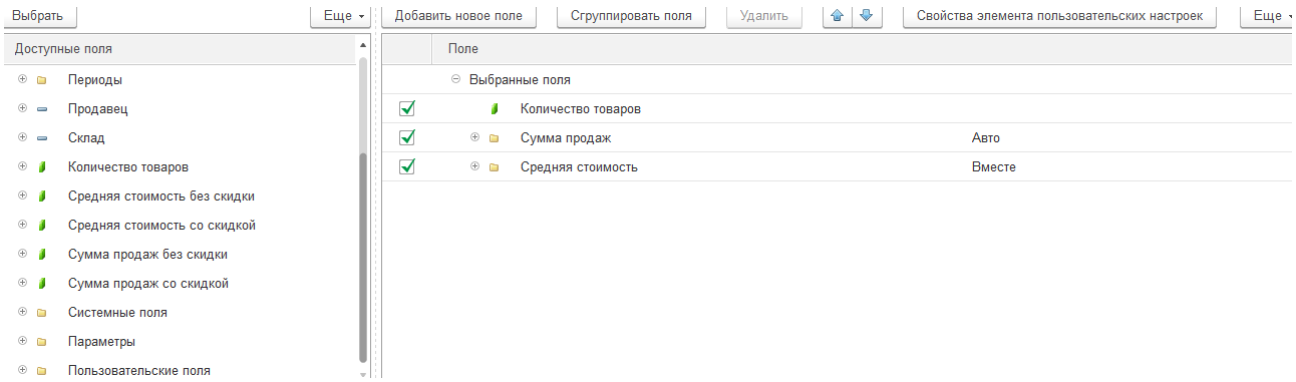

Рисунок 5.10 – Раздел «Поля»

– Отбор. Содержит отборы, используемые в варианте отчета (рис. 5.11).

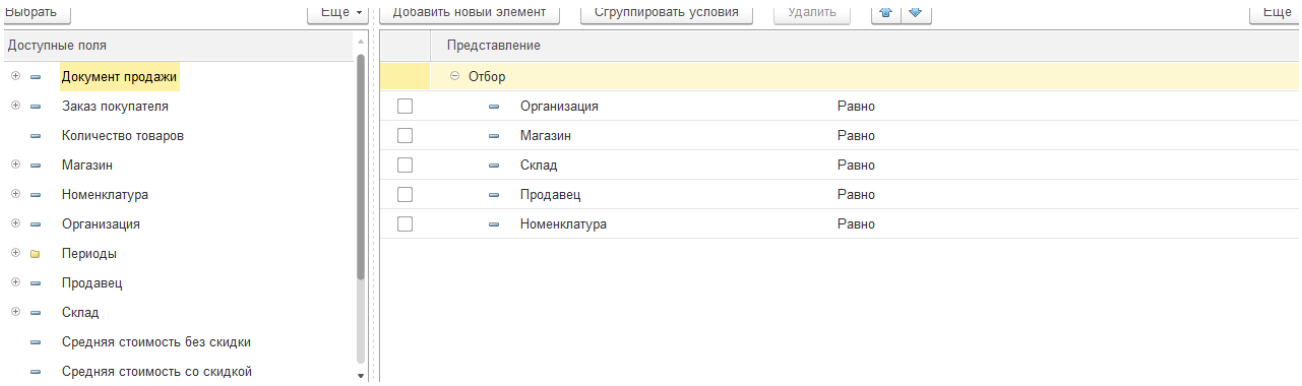

Рисунок 5.11 – Раздел «Отбор»

– Сортировка. Содержит поля сортировки, используемые в варианте отчета (рис. 5.12).

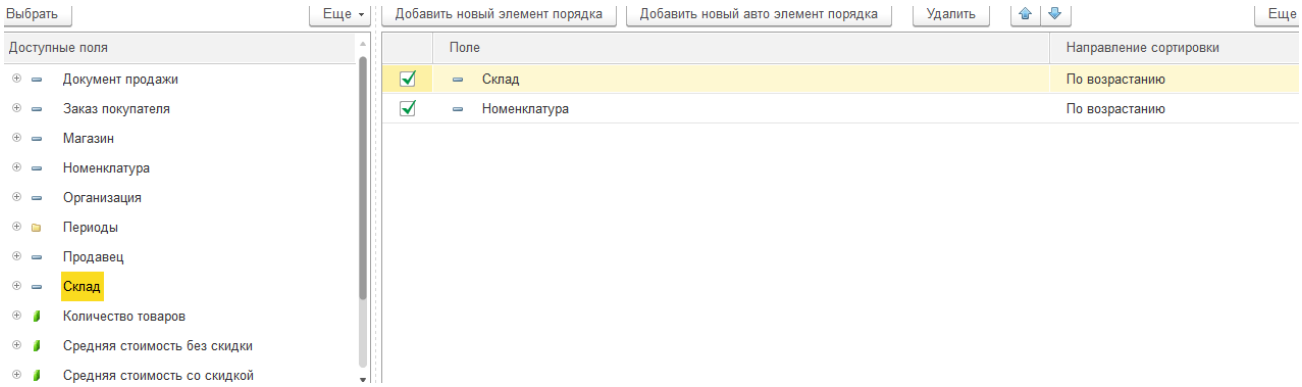

Рисунок 5.12 – Раздел «Сортировка»

– Условное оформление. Содержит элементы условного оформления, используемые в варианте отчета (рис. 5.13).

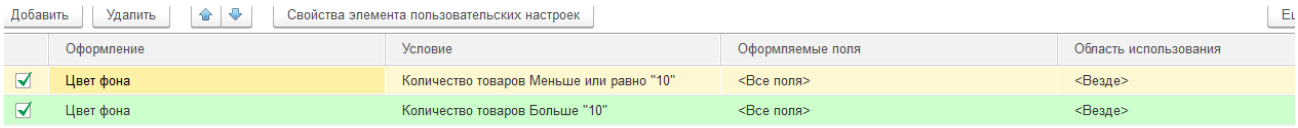

Рисунок 5.13 – Раздел «Условное форматирование»

– Дополнительные настройки. Содержит дополнительные настройки оформления отчета (рис. 5.14).

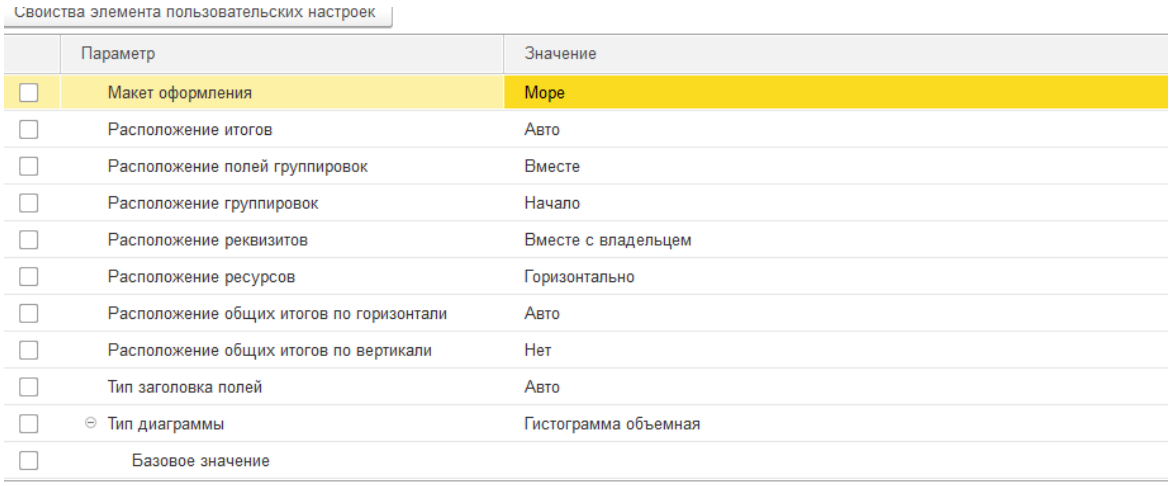

Рисунок 5.14 – Раздел «Дополнительные настройки»

Варианты заданий

Задание для всех вариантов: создать в созданной ранее базе данных не менее трех отчетов, используя запросы, после чего изменить их, используя пользовательскую настройку отчетов.

ЛАБОРАТОРНАЯ РАБОТА № 6. НАПИСАНИЕ ЗАПРОСОВ, РАЗРАБОТКА ОТЧЕТОВ С ПОМОЩЬЮ СИСТЕМЫ КОМПОНОВКИ ДАННЫХ

Цель лабораторной работы

Получение практических навыков в написании запросов и разработки отчетов с помощью системы компоновки данных.

Методические указания

Язык запросов в 1С, начиная с версии 8, сегодня стал полезным инструментом для работы с базами данных, который позволяет читать из них, но не записывать. Синтаксически язык запросов очень схож с языком SQL, но на русском языке (табл. 6.1).

Таблица 6.1 – Сопоставление операторов

Операторы языка запросов 1С	Оператор SQL
ВЫБРАТЬ	SELECT
ГДЕ	WHERE
РАЗЛИЧНЫЕ	DISTINCT
ПОДОБНО	LIKE
ПЕРВЫЕ	TOP
ВЫБОР	CASE
СОЕДИНЕНИЕ	JOIN
СГРУППИРОВАТЬ ПО	GROUP BY
ОБЪЕДИНИТЬ	UNION
УПОРЯДОЧИТЬ ПО	ORDER BY
ИМЕЮЩИЕ	HAVING
ИЗ	FROM
В	IN

Также в языке запросов 1С существуют операторы, аналогов которых нет в SQL, например:

– В ИЕРАРХИИ;

– ПОМЕСТИТЬ;

– ИНДЕКСИРОВАТЬ ПО.

Оператор В ИЕРАРХИИ позволяет выбрать все элементы иерархического справочника, которые входят в иерархию переданной ссылки (рис. 6.1).

```
ВЫБРАТЬ
   Товары.Ссылка,
   Товары.Артикул
ИЗ
   Справочник.Товары КАК Товары
ГДЕ
   Товары.Ссылка В ИЕРАРХИИ(&Цитрусовые)"
```

Рисунок 6.1 – Пример использования оператора В ИЕРАРХИИ

Оператор ПОМЕСТИТЬ помещает результат во временную таблицу (рис. 6.2).

```
ВЫБРАТЬ
  Пользователи.Ссылка КАК Ссылка,
  Пользователи.Родитель КАК Родитель,
  Пользователи.Наименование КАК Наименование
ПОМЕСТИТЬ ОтобранныеПользователи
ИЗ
  Справочник.Пользователи КАК Пользователи
ГДЕ
  Пользователи.Ссылка = &Справочник;
ВЫБРАТЬ
  ОтобранныеПользователи.Ссылка КАК Ссылка,
  ОтобранныеПользователи.Родитель КАК Родитель,
  ОтобранныеПользователи.Наименование КАК Наименование
ИЗ
  ОтобранныеПользователи КАК ОтобранныеПользователи
```

Рисунок 6.2 – Пример использования оператора ПОМЕСТИТЬ

Оператор ИНДЕКСИРОВАТЬ ПО применяется совместно с оператором ПОМЕСТИТЬ. При создании временной таблицы этим оператором можно проиндексировать создаваемую таблицу, что существенно ускоряет работу с ней (но только если индекс подходит под запрос).

Запрос можно писать как вручную, так и используя конструктор.

После написания запроса, СКД автоматически заполнит таблицу с настройками полей (рис. 6.3).

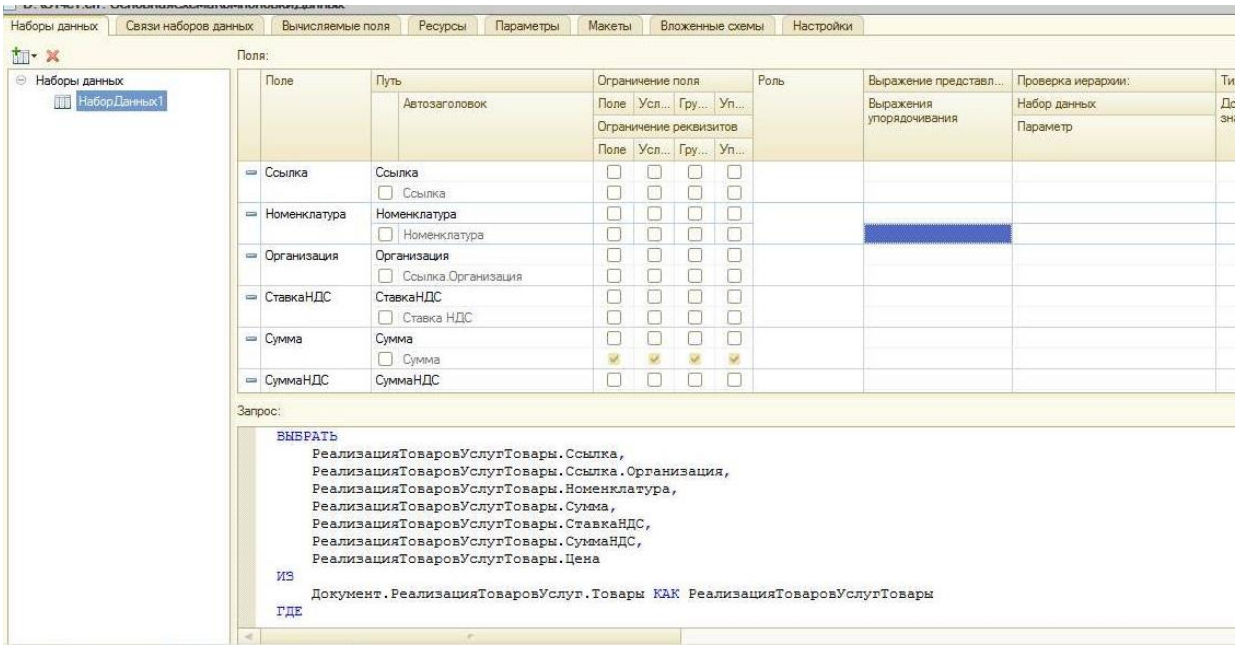

Рисунок 6.3 – Таблица СКД

Настройки:

✓ Автозаголовок: можно задать произвольный заголовок для поля, он будет выводиться, если установить флажок.

✓ Выражение представления: определяет представление поля, выводимого в отчет.

✓ Тип значения: здесь можно указать тип поля. Это полезно, например, при создании отбора по данному полю – пользователю не придется самому выбирать тип данных для значения отбора.

✓ Оформление: позволяет отформатировать выводимые данные. Здесь можно задать цвет, шрифт, размер, ширину поля и т. п.

Под полями-ресурсами в системе компоновки данных подразумеваются поля, значения которых рассчитываются на основании детальных записей, входящих в группировку. По сути, ресурсы являются групповыми или общими итогами отчета. Итоги по ресурсам можно рассчитывать при помощи функций языка выражений СКД, самые простые из которых Сумма(), Среднее(), Максимум(), Минимум() и Количество().

Чтобы задать ресурсы отчета надо перейти на закладку «Ресурсы» и перетащить необходимые поля отчета в таблицу ресурсов. После этого необходимо задать выражение, также вы можете выбрать группировки, для которых хотите видеть итоги по данному ресурсу, это можно сделать в столбце «Рассчитывать по…» (рис. 6.4).

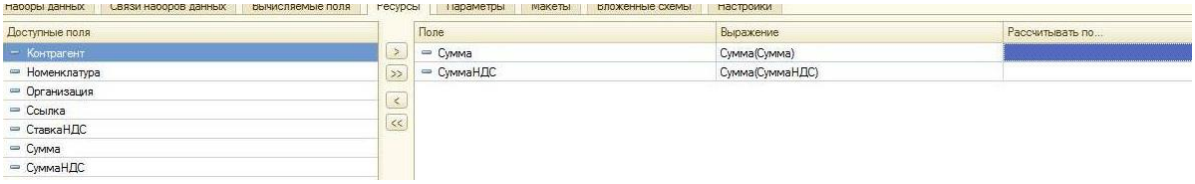

Рисунок 6.4 – Вкладка «Ресурсы»

Все параметры, указанные в запросе, отобразятся на закладке «Параметры схемы компоновки»:

✓ Столбцы «Имя», «Заголовок» и «Тип» заполняются автоматически, и менять их значения не следует без необходимости;

✓ Доступен список значений. Устанавливается, если нужно передать в параметр список, иначе туда попадет только первый элемент списка;

✓ Значение. Здесь можно указать значение по умолчанию для параметра. В нашем примере выберем для параметра «Организации» значение «Элементы» (пустая ссылка на справочник Организации);

✓ Включать в доступные поля. Если снять этот флаг, параметра не будет видно в настройках: в выбранных полях, отборе;

✓ Ограничение доступности. Флаг отвечает за возможность установки значения параметра в настройке СКД (рис. 6.5).

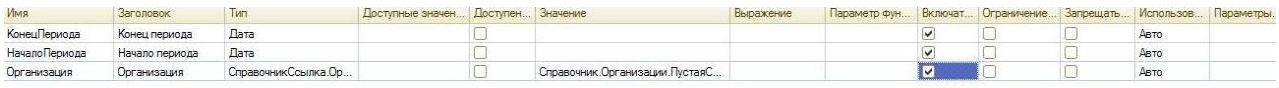

Рисунок 6.5 – Вкладка параметры

Варианты заданий

Задание для всех вариантов: добавить в созданную ранее базу данных не менее трех запросов с использованием операторов ГДЕ, вручную изменить их, используя систему компоновки данных.

ЛАБОРАТОРНАЯ РАБОТА № 7. РАБОТА С УПРАВЛЯЕМЫМИ ФОРМАМИ ОБЪЕКТОВ

Цель лабораторной работы

Получение практических навыков в работе с управляемыми формами объектов.

Методические указания

Формы предназначены для отображения и редактирования информации, содержащейся в базе данных. Формы могут принадлежать конкретным объектам конфигурации или существовать отдельно от них и использоваться всем прикладным решением в целом.

Например, справочник может иметь несколько форм, которые будут использоваться для определенных целей — редактирования элемента справочника, отображения списка и т. д.

Каждый объект конфигурации может использоваться для выполнения некоторых стандартных действий. Например, для любого справочника может потребоваться отображать список его элементов, отображать отдельные элементы справочника, отображать группу справочника, выбирать элементы и группы элементов из справочника (рис. 7.1). Для любого документа список таких действий будет гораздо меньше: просмотр списка документов, выбор из списка документов и просмотр отдельного документа (рис. 7.2).

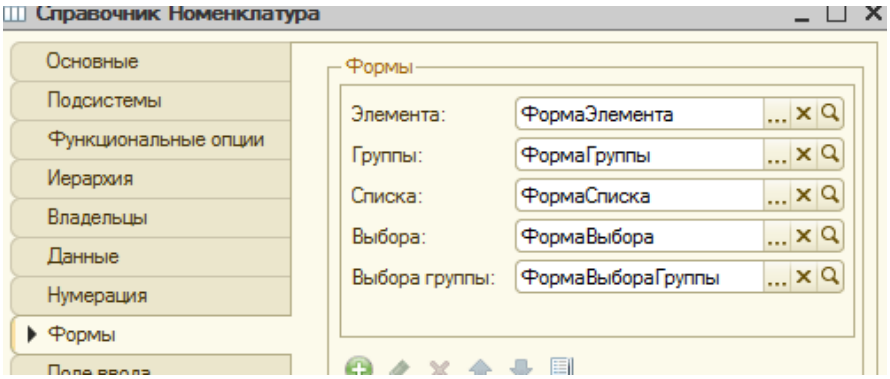

Рисунок 7.1 – Вкладка формы справочника

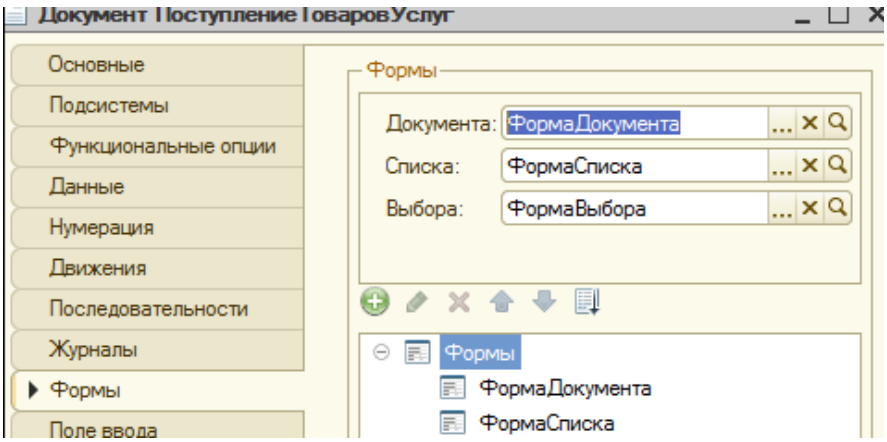

Рисунок 7.2 – Вкладка формы документа

Отображаемая часть формы (видимая пользователю) описывается как дерево, включающее элементы формы (рис. 7.3).

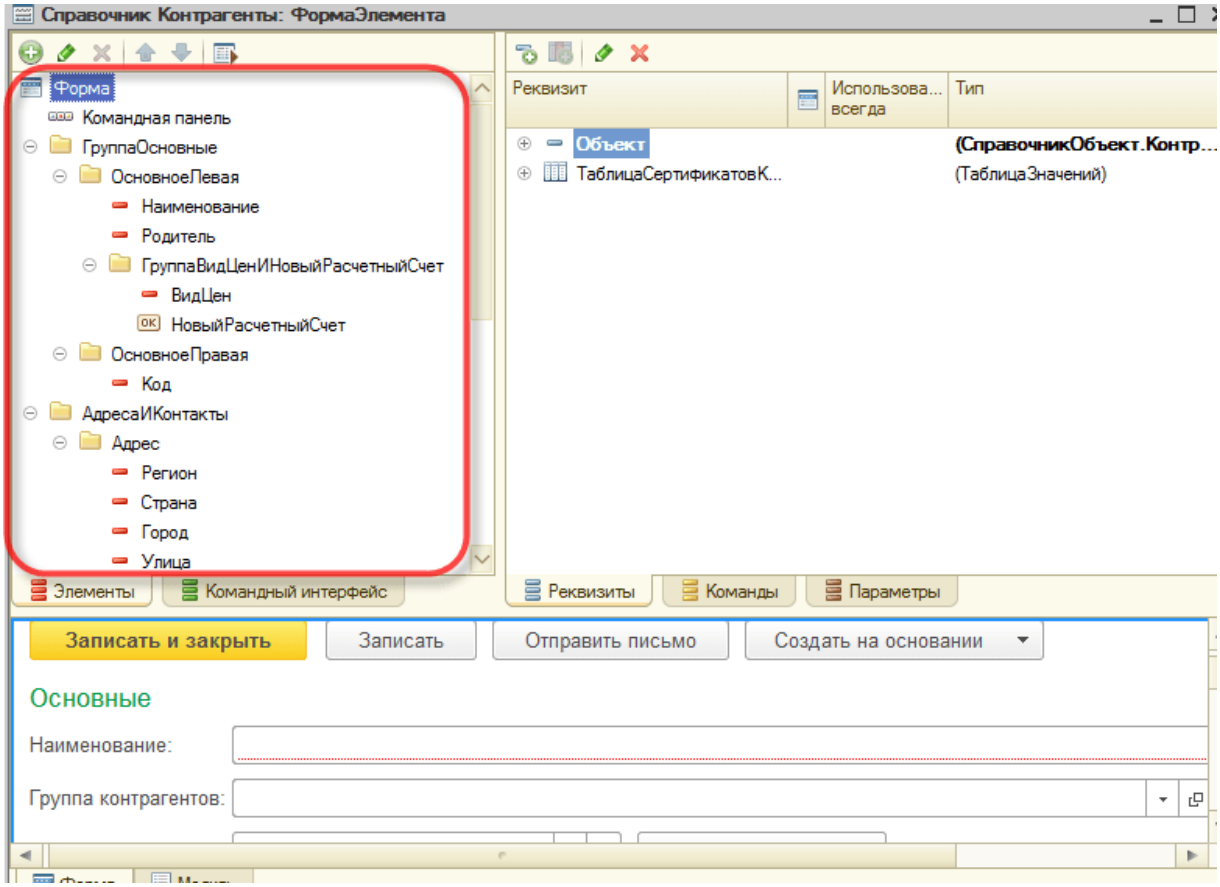

Рисунок 7.3 – Редактор форм

Элементы могут представлять собой поля ввода, флажки, переключатели, кнопки и т. д. Кроме того, элемент может быть группой, включающей другие элементы. Группа может представляться как панель с рамкой, панель со страницами (закладками), собственно страница, командная панель. Помимо этого, элемент может представлять собой таблицу, которая тоже включает элементы (колонки). Структура элементов описывает то, как будет выглядеть форма.

Вся функциональность формы описывается в виде реквизитов и команд. Реквизиты – это данные, с которыми работает форма, а команды – выполняемые действия. Таким образом, разработчик в редакторе формы должен включить в форму необходимые реквизиты и команды, создать отображающие их элементы формы и, если необходимо, скомпоновать элементы в группы (рис. 7.4).

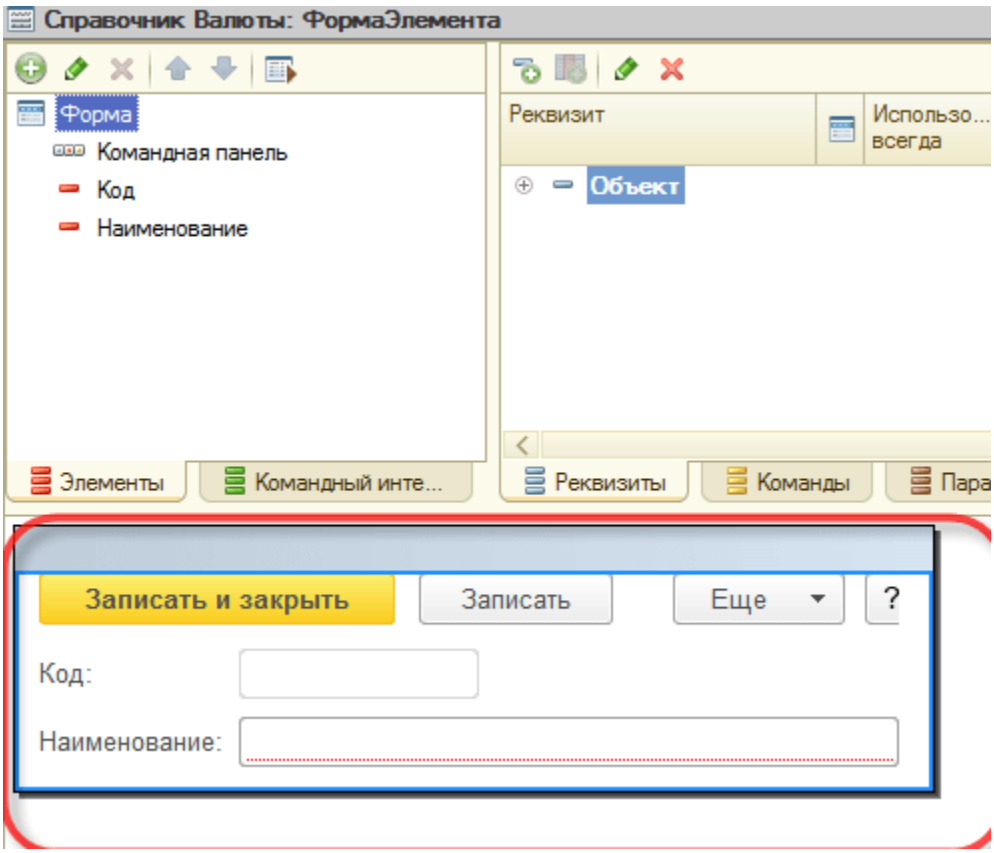

Рисунок 7.4 – Пример простой формы

Разработчик может влиять на расположение элементов различными установками. Он может определять порядок элементов, указывать желаемую ширину и высоту. Однако это является только некоторой дополнительной информацией, помогающей системе отобразить форму.

В формах разработчик может использовать не только команды самой формы, но и глобальные команды, используемые в командном интерфейсе всей конфигурации. Кроме того, реализована возможность создания параметризуемых команд, которые будут открывать другие формы с учетом конкретных данных текущей формы.

Варианты заданий

Задание для всех вариантов: для всех созданных ранее справочников изменить формы (цвет, шрифт, расположение элементов).

ЛАБОРАТОРНАЯ РАБОТА № 8. НАПИСАНИЕ ОБРАБОТЧИКА СОБЫТИЙ ДЛЯ ДОКУМЕНТА

Цель лабораторной работы

Получение практических навыков в написании обработчика событий документа.

Методические указания

В ходе решения различных задач пользователей иногда возникает необходимость уже сформированные движения документов (а именно определенные наборы регистров) подвергать какой-либо корректировке.

Для данных целей очень хорошо подходит объект «Обработчик событий», который позволяет выполнять какие-то действия при наступлении определенного события для большого количества объектов (например, при записи платежных документов или при установке нового номера справочников, связанных с налоговым учетом).

Источник – это объект (например, документ или список документов) для которого будет вызываться действия.

Событие – само действие, после которого будет выполняться код.

Обработчик – указание на процедуру, в которой будет происходить обработка.

Во многих обработчиках есть параметр «Отказ». Там, где этот параметр присутствует, это означает, что в этом обработчике еще можно отказаться от записи, присвоив параметру «Отказ» значение «Истина», и тогда запись произведена не будет. Важно: при программной записи события модуля формы не запускаются.

Последовательность запуска событий:

1 Модуль формы «ПередЗаписью(Отказ, ПараметрыЗаписи)».

Выполняется на клиенте! Этот обработчик следует использовать, если необходимо организовать диалог с пользователем перед тем, как записать объект. Запросить дополнительную информацию, предупредить о чём-либо, дать возможность отказаться и т. п.

2 Модуль формы «ОбработкаПроверкиЗаполненияНаСервере(Отказ, ПроверяемыеРеквизиты)».

3 Модуль объекта «ОбработкаПроверкиЗаполнения (Отказ, ПроверяемыеРеквизиты)».

Сначала вызывается событие формы «ОбработкаПроверкиЗаполненияНаСервере». На данном этапе есть доступ к данным формы. После этого в памяти сервера создается прикладной объект, соответствующий типу основного реквизита формы, и его данные заполняются из данных формы.

4 Модуль формы «ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)».

В этом обработчике можно дозаполнять реквизиты объекта (через параметр «ТекущийОбъект») или провести дополнительные проверки. Есть доступ к данным формы.

ТекущийОбъект – это объект, который реально будет записан в информационную базу и дозаполнять реквизиты нужно именно через параметр «ТекущийОбъект».

5 Модуль объекта «ПередЗаписью(Отказ)».

В этом обработчике можно провести дополнительные проверки и отказаться от записи. Для документов в параметры данного обработчика добавляются ещё два параметра: «РежимЗаписи», «РежимПроведения».

6 Модуль объекта «ПриУстановкеНовогоНомера(СтандартнаяОбработка, Префикс)».

Возникает в момент, когда выполняется установка номера нового документа, задачи или бизнес-процесса.

7 Модуль объекта «ОбработкаУдаленияПроведения (Отказ)».

Этот обработчик запускается только при отмене проведения документов с целью удаления движений из регистров. При этом неважно как отменяется проведение документа - программно или интерактивно.

8 Модуль объекта «ПриЗаписи(Отказ)».

Вызывается после записи объекта в базу данных, но до окончания транзакции записи. Назначение этого обработчика – записать в базу данных дополнительную информацию, связанную с данными записываемого объекта.

9 Модуль объекта «ОбработкаПроведения (Отказ, РежимПроведения)».

Этот обработчик запускается только при проведении документов. При этом неважно как проводится документ – программно или интерактивно. Основное назначение процедуры-обработчика данного события – генерация движений по документу. Именно в данном обработчике прописываются алгоритмы записей в регистры, т. е. программно формируются движения документа.

10 Модуль формы «ПриЗаписиНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)».

Вызывается после записи объекта в базу данных, но до окончания транзакции записи. Есть последний шанс отказаться от записи. Назначение этого обработчика – записать в базу данных дополнительную информацию, связанную с данными записываемого объекта.

11 Модуль формы «ПослеЗаписиНаСервере(ТекущийОбъект, ПараметрыЗаписи)».

После завершения транзакции записи выполняется преобразование данных записанного объекта в данные формы. После этого вызывается данный обработчик. Это последний обработчик, в котором по отдельности доступны данные формы и объект, который был записан.

12 Модуль формы «ПослеЗаписи(ПараметрыЗаписи) «

Выполняется на клиенте! Можно использовать для того, чтобы визуально что-то отобразить на форме или организовать диалог с пользователем,

например выдать предупреждение. Для создания обработчика событий нужно открыть «Модуль формы2. Далее перейти на закладку «Модуль» редактора формы (рис. 8.1).

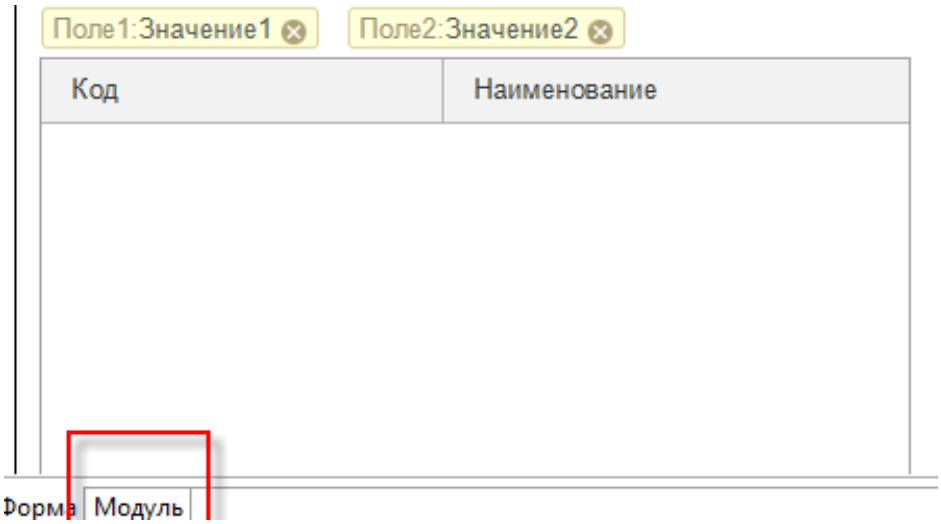

Рисунок 8.1 – Вкладка «Модуль»

Пустой модуль откроется в редакторе встроенного языка.

Далее нужно выбрать пункт добавить обработчик события в контекстном меню (рис. 8.2).

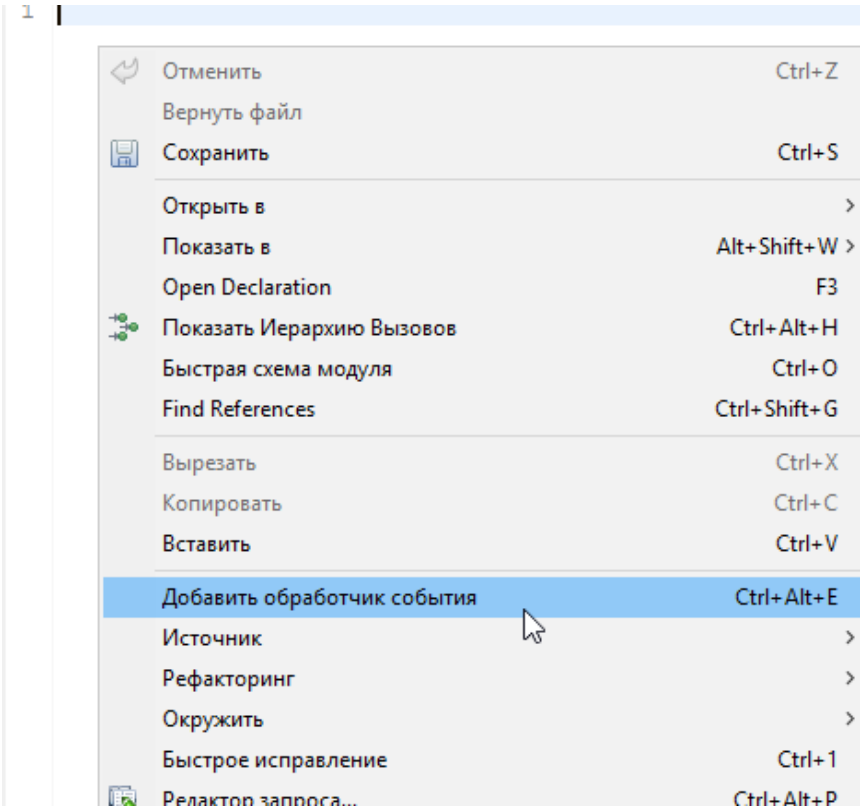

Рисунок 8.2 – Добавление обработчика событий

Далее нужно выбрать событие (рис. 8.3).

Далее появится часть кода, где и будет писаться код для обработчика событий (рис. 8.4).

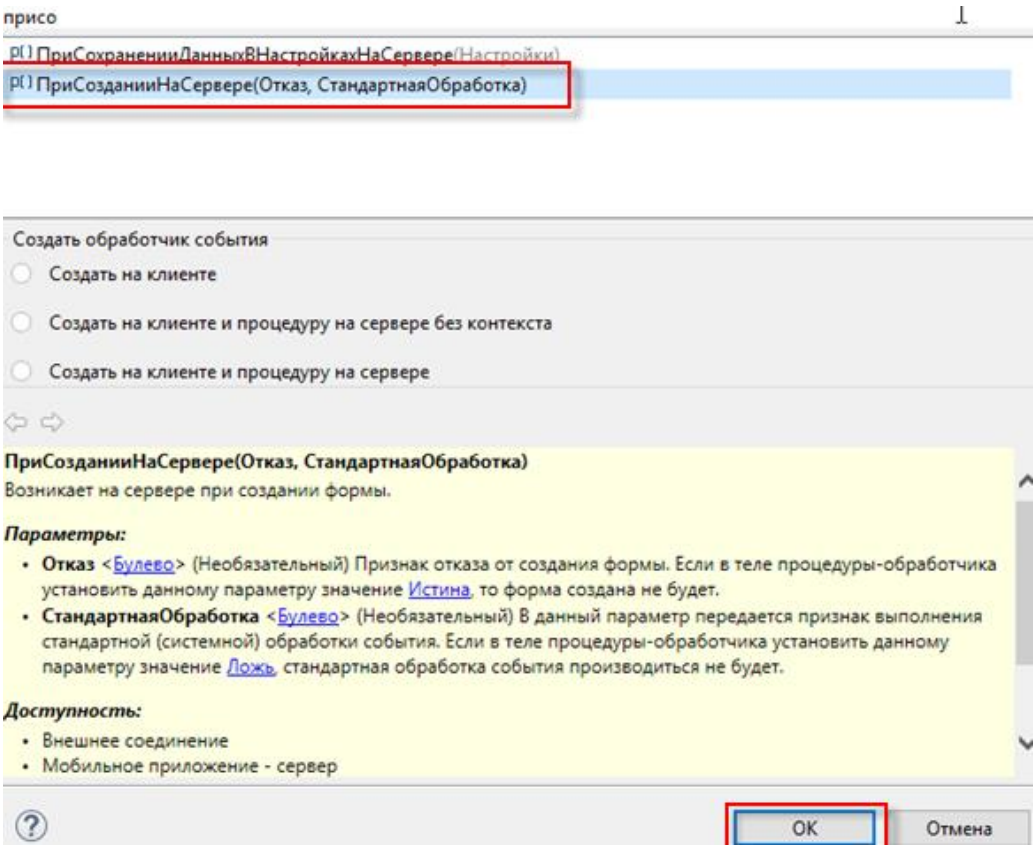

Рисунок 8.3 – Выбор событий

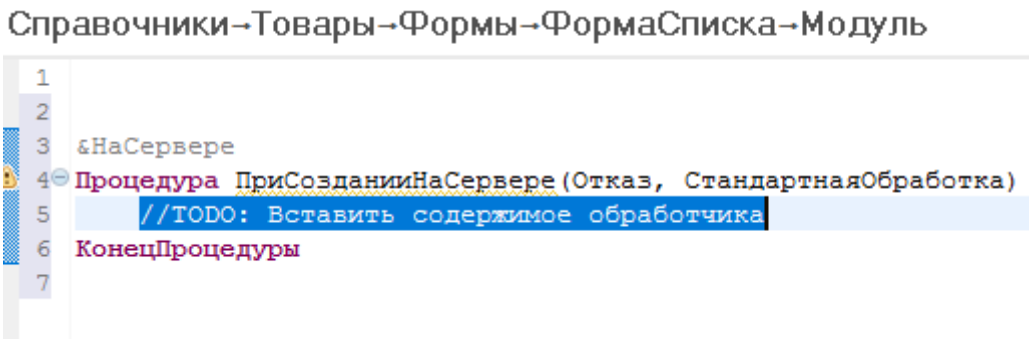

Рисунок 8.4 – Процедура обработчика событий

Варианты заданий

Задание для всех вариантов: добавить в созданную ранее базу данных не менее трех обработчиков событий для проверки правильности ввода данных (в случае ошибки должно появляться сообщение об отказе).

ЛАБОРАТОРНАЯ РАБОТА № 9. НАПИСАНИЕ КОДА НА ВСТРОЕННОМ ЯЗЫКЕ РАЗРАБОТКИ

Цель лабораторной работы

Получение практических навыков в написании кода на встроенном языке разработки.

Методические указания

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.

Переменные – это место хранения значений. Переменная может называться любым словом (рис. 9.1).

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

Рисунок 9.1 – Пример использования переменных

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()) (рис. 9.2).

```
Чтото = Новый Массив;
Чтото.Добавить(22); //в ячейке 1 значение 22
Чтото.Добавить(33); //в ячейке 2 значение 33
Сообщить(Чтото[0]); //выводим значение 1й ячейки
```

Рисунок 9.2 – Пример использования массива

«СписокЗначений». То же, что и массив, только к каждой ячейке можно подписать комментарий (рис. 9.3).

```
Чтото = Новый СписокЗначений;
Чтото.Добавить(22, «Это 22»); //в ячейке 1 значение 22
Чтото.Добавить(33, «Это 33»); //в ячейке 2 значение 33
Сообщить(Чтото.Получить(0).Значение); //выводим значение 1й ячейки
//а могли бы вывести и комментарий, тогда бы написали .Представление а не .Значение
```

Рисунок 9.3 – Пример использования списка значений

Также существует «ТаблицаЗначений», которая представляет собой двумерный массив.

Строки:

– 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 «|»;

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

Операторы:

– Условие. Определенные строки кода будут выполнены, если будет выполнено условие (рис. 9.4).

```
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;
```

Рисунок 9.4 – Пример использования условия

– Цикл. Определенные строки кода будут выполнены указанное количество раз (рис. 9.5).

```
//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
```

Рисунок 9.5 – Пример использования цикла

Все объекты, с которыми производится работа в языке, являются полноценными объектами, то есть могут иметь свои данные и свои методы.

Объекты языка создаются с помощью команды «Новый».

Объекты 1С нельзя создать – доступ к ним можно получить с помощью, так называемых менеджеров, названных соответственно по веткам конфигурации: Справочники, Документы и т. п. (рис. 9.6):

```
Справочники.Номенклатура.СоздатьЭлемент();
Документы.Накладная.НайтиПоНомеру(«…»);
```

Рисунок 9.6 – Пример использования объектов

Функции и Процедуры – это способ взять несколько строчек кода и назвать их каким-то словом, как переменную. Если потом написать ее название в другом месте – будет вызвана эта функция (то есть выполнены эти строки кода) (рис. 9.7).

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

//Вызываем
КакоетоДействие();
```

Рисунок 9.7 – Пример использования процедуры

Функция отличается от процедуры тем, что она может вернуть значение, которое было вычислено в результате выполнения этих строк кода (Рисунок 9.8):

```
Функция Посчитать(Переменная1, Переменная2)
   Возврат Переменная1 + Переменная2;
КонецФункции

//Вызываем
Чтото = Посчитать(12, 20);
Сообщить(Чтото); //будет сообщение «32»
```

Рисунок 9.8 – Пример использования функции

Варианты заданий

Задание для всех вариантов: написать код в созданных в предыдущей лабораторной работе обработчиках событий.

ЛАБОРАТОРНАЯ РАБОТА № 10. ПРОГРАММИРОВАНИЕ ФОРМ

Цель лабораторной работы

Получение практических навыков в программировании форм.

Методические указания

Переопределить обработчики событий формы можно в обработчике «ПриСозданииНаСервере» (кроме самого обрабочика «ПриСозданииНаСервере») или в теле модуля формы в контексте сервера (например, «Переопределение событий ПриЧтенииинаСервере» и «ПриСозданииНаСервере») (рис. 10.1).

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    ЭтаФорма.УстановитьДействие(“ОбработкаВыбора”, “пр_ОбработкаВыбора”)
КонецПроцедуры
```

Рисунок 10.1 – Переопределение «ПриСозданииНаСервере»

В теле модуля формы (рис. 10.2).

```
#Если Сервер Тогда
    ЭтаФорма.УстановитьДействие("ПриСозданииНаСервере", "пр_ПриСозданииНаСервере");
#КонецЕсли
```

Рисунок 10.2 – Переопределение в теле модуля формы

В новой процедуре нужно добавить вызов основной, если такая процедура есть (рис. 10.3).

```
&НаСервере
Процедура пр_ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    ПриСозданииНаСервере(Отказ, СтандартнаяОбработка);
КонецПроцедуры
```

Рисунок 10.3 – Переопределение в новой процедуре

Программное создание групп формы (рис. 10.4).

```
&НаСервере
Процедура пр_СоздатьГруппы()

    ГруппаСтраницы =  Элементы.Добавить("пр_Страницы", Тип("ГруппаФормы"),ЭтаФорма);
    ГруппаСтраницы.Вид = ВидГруппыФормы.Страницы;

    НоваяСтраница = Элементы.Добавить("пр_Страница", Тип("ГруппаФормы"),ГруппаСтраницы);
    НоваяСтраница.Вид = ВидГруппыФормы.Страница;
    НоваяСтраница.Заголовок = "Страница 1";

    НоваяГруппа = Элементы.Добавить("пр_Группа1", Тип("ГруппаФормы"),НоваяСтраница);
    НоваяГруппа.Вид = ВидГруппыФормы.ОбычнаяГруппа;
    НоваяГруппа.Группировка = ГруппировкаПодчиненныхЭлементовФормы.ГоризонтальнаяЕслиВозможно;
    НоваяГруппа.Заголовок = "Группа 1";

КонецПроцедуры
```

Рисунок 10.4 – Пример создания группы

Удалить команду можно при помощи метода коллекции формы команд: «Команды.Удалить(<Команда>)». Удалять можно только те команды, которые были созданы программно.

Программное добавление команды на форму (рис. 10.5).

```
&НаСервере
Процедура пр_СоздатьНовуюКоманду()

    //создать новую команду у формы
    НоваяКоманда                = Команды.Добавить("пр_Команда1");
    НоваяКоманда.Действие     = "пр_Команда1";
    НоваяКоманда.Картинка     = БиблиотекаКартинок.Облако;
    НоваяКоманда.Отображение = ОтображениеКнопки.Картинка;

    //вывести команду в элементы
    ЭлементКоманда                      = Элементы.Добавить("пр_Команда1", Тип("КнопкаФормы"), Элементы.ФормаКоманднаяПанель);
    ЭлементКоманда.Заголовок     = "Вывести сообщение";
    ЭлементКоманда.ИмяКоманды     = "пр_Команда1";

    //удалить команду
    //Команды.Удалить(НоваяКоманда);

КонецПроцедуры

&НаКлиенте
Процедура пр_Команда1(Команда)

    Сообщить("Команда выполнена.");

КонецПроцедуры
```

Рисунок 10.5 – Добавление команды на форму

Программное создание декораций форм (рис. 10.6, 10.7).

```
Элемент                     = Элементы.Добавить("Надпись1", Тип("ДекорацияФормы"), Элементы["пр_Группа1"]);
Элемент.Вид                 = ВидДекорацииФормы.Надпись;
Элемент.Заголовок           = "Добавленная надпись";
```

Рисунок 10.6 – Добавление надписи

```
Элемент = Элементы.Добавить("Картинка1", Тип("ДекорацияФормы"), Элементы["пр_Группа1"]);
Элемент.Вид = ВидДекорацииФормы.Картинка;
Элемент.Картинка = БиблиотекаКартинок.Бесконечность;
```

Рисунок 10.7 – Добавление картинки

Добавлять новые реквизиты в управляемую форму и удалять необходимо с помощью метода ИзменитьРеквизиты(), куда в параметры передается массив добавляемых реквизитов и массив удаляемых. Удалять при этом можно только те реквизиты, которые были созданы программно (рис. 10.8).

```
&НаСервере
Процедура пр_УдалитьРеквизиты()

        // Массив для удаляемых реквизитов
        УдаляемыеРеквизиты     = Новый Массив;

    //указываем путь к удаляемому реквизиту
        УдаляемыеРеквизиты.Добавить("пр_УдалитьКоличество");
    //К добавленным реквизитам нужно обращаться через переменную ЭтаФорма

        // Добавим новые реквизиты в форму
        ИзменитьРеквизиты(,УдаляемыеРеквизиты);

КонецПроцедуры
```

Рисунок 10.8 – Удаление реквизитов

Программное добавление реквизита на форму (рис. 10.9).

```
&НаСервере
Процедура пр_СоздатьРеквизиты()

        // Массив для новых реквизитов
        ДобавляемыеРеквизиты     = Новый Массив;

        // Опишем ревизиты формы
        Реквизит_Использование = Новый РеквизитФормы("пр_Использование",     Новый ОписаниеТипов("Булево"), "", "Использование");
        Реквизит_Номенклатура = Новый РеквизитФормы("пр_Номенклатура",     Новый ОписаниеТипов("СправочникСсылка.Номенклатура"), "", "Номенклатура");
        Реквизит_Характеристика = Новый РеквизитФормы("пр_Характеристика",     Новый ОписаниеТипов("СправочникСсылка.ХарактеристикиНоменклатуры"), "", "Характеристика");
         Реквизит_Количество = Новый РеквизитФормы("пр_Количество",     Новый ОписаниеТипов("Число", , , Новый КвалификаторыЧисла(10, 3)), "", "Количество");
         Реквизит_УдалитьКоличество = Новый РеквизитФормы("пр_УдалитьКоличество",     Новый ОписаниеТипов("Число", , , Новый КвалификаторыЧисла(10, 3)), "", "Удалить_Количество");
       //если используется БСП, то можно для определения описания типов использовать функцию
       //ОбщегоНазначения.ОписаниеТипаСтрока(ДлинаСтроки)
       //ОбщегоНазначения.ОписаниеТипаЧисло(Разрядность, РазрядностьДробнойЧасти = 0, ЗнакЧисла = Неопределено)
       //ОбщегоНазначения.ОписаниеТипаДата(ЧастиДаты)
         Реквизит_Информация = Новый РеквизитФормы("пр_Информация",     ОбщегоНазначения.ОписаниеТипаСтрока(100), "", "Информация");

       // Для наглядности заполним массив после описания реквизитов формы
        ДобавляемыеРеквизиты.Добавить(Реквизит_Использование);
        ДобавляемыеРеквизиты.Добавить(Реквизит_Номенклатура);
        ДобавляемыеРеквизиты.Добавить(Реквизит_Характеристика);
        ДобавляемыеРеквизиты.Добавить(Реквизит_Количество);
        ДобавляемыеРеквизиты.Добавить(Реквизит_УдалитьКоличество);
        ДобавляемыеРеквизиты.Добавить(Реквизит_Информация);

        // Добавим новые реквизиты в форму
        ИзменитьРеквизиты(ДобавляемыеРеквизиты);

КонецПроцедуры
```

Рисунок 10.9 – Добавление реквизитов

Вывод реквизитов на форму (рис. 10.10).

```
&НаСервере
Процедура пр_ВывестиРеквизитыНаФорму()

    НовыйЭлемент = Элементы.Добавить("пр_Использование",
Тип("ПолеФормы"), Элементы.пр_Страница1);
    НовыйЭлемент.ПутьКДанным = "пр_Использование";
    НовыйЭлемент.Вид = ВидПоляФор-
мы.ПолеФлажка;
    НовыйЭлемент.ПоложениеЗаголовка = ПоложениеЗаголов-
каЭлементаФормы.Право;
    //установим обработчик события элементы
    НовыйЭлемент.УстановитьДействие("ПриИзменении",
"пр_ИспользованиеПриИзменении");
```

Рисунок 10.10 – Вывод реквизитов на форму (*начало*)

```
        НовыйЭлемент = Элементы.Добавить("пр_Номенклатура",
Тип("ПолеФормы"), Элементы.пр_Страница1);
        НовыйЭлемент.ПутьКДанным = "пр_Номенклатура";
        НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;
        //установить параметр выбора элемента
        НовыйМассив = Новый Массив();
        НовыйПараметр = Новый ПараметрВыбо-
ра("Отбор.ПометкаУдаления", Ложь);
        НовыйМассив.Добавить(НовыйПараметр);
        НовыеПараметры = Новый ФиксированныйМассив(НовыйМассив);
        НовыйЭлемент.ПараметрыВыбора = НовыеПараметры;
        НовыйЭлемент.УстановитьДействие("ПриИзменении",
"пр_НоменклатураПриИзменении");

        НовыйЭлемент = Элементы.Добавить("пр_Характеристика",
Тип("ПолеФормы"), Элементы.пр_Страница1);
        НовыйЭлемент.ПутьКДанным = "пр_Характеристика";
        НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;
        НовыйЭлемент.Заголовок = "Характеристика номенкла-
туры";
        //добавить связь параметров выбора по реквизиту владелец в зависи-
мости от выбранной номенклатуры
        НоваяСвязь = Новый СвязьПараметраВыбора("Отбор.Владелец",
"пр_Номенклатура");
        НовыйМассив = Новый Массив();
        НовыйМассив.Добавить(НоваяСвязь);
        НовыйЭлемент.СвязиПараметровВыбора = Новый Фиксированный-
Массив(НовыйМассив);
        НовыйЭлемент.УстановитьДействие("ПриИзменении",
"пр_ХарактеристикаПриИзменении");

        НовыйЭлемент = Элементы.Добавить("пр_Количество",
Тип("ПолеФормы"), Элементы.пр_Страница1);
        НовыйЭлемент.ПутьКДанным = "пр_Количество";
        НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;
        //установить формат
        НовыйЭлемент.ФорматРедактирования = "ЧДЦ=0; ЧРГ=.;
ЧН=Отсутствует";

        НовыйЭлемент = Элементы.Добавить("пр_Информация",
Тип("ПолеФормы"), Элементы.пр_Страница1);
        НовыйЭлемент.ПутьКДанным = "пр_Информация";
        НовыйЭлемент.Вид = ВидПоляФормы.ПолеВвода;
```

Рисунок 10.10 – Вывод реквизитов на форму (*продолжение*)

```
        //установить список выбора
        НовыйЭлемент.РежимВыбораИзСписка          = Истина;
        Массив = Новый Массив;
        Массив.Добавить("Необходимо дозаказать");
        Массив.Добавить("Достаточно");
        Массив.Добавить("Оформить полный заказ");
        НовыйЭлемент.СписокВыбора.ЗагрузитьЗначения(Массив);

    КонецПроцедуры

    &НаКлиенте
    Процедура пр_ИспользованиеПриИзменении(Элемент)
    КонецПроцедуры
```

Рисунок 10.10 – Вывод реквизитов на форму (*окончание*)

После описания всех процедур выводим их в процедуре «ПриСозданиНаСервере». Также можно изменить программно некоторые свойства формы, например, заголовок. Чтобы был виден только наш заголовок, нужно отключить свойство Автозаголовок (рис. 10.11).

```
&НаСервере

&НаСервере
Процедура пр_ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

        //установим заголовок формы
        ЭтаФорма.Заголовок              = "Шаблон для программной работы с реквизитами, командами и элементами формы";
        ЭтаФорма.АвтоЗаголовок  = Ложь;

        пр_СоздатьГруппы();
        пр_СоздатьНовуюКоманду();
        пр_СоздатьДекорацию();
        пр_СоздатьРеквизиты();
        пр_УдалитьРеквизиты();
        пр_ВывестиРеквизитыНаФорму();

КонецПроцедуры
```

Рисунок 10.11 – Вызов процедур на сервере

Варианты заданий

Задание для всех вариантов: изменить программно не менее трех любых форм в базе данных, созданной в предыдущих лабораторных работах.

ЛАБОРАТОРНАЯ РАБОТА № 11. ПРОГРАММНАЯ ОБРАБОТКА ДАННЫХ

Цель лабораторной работы

Получение практических навыков в программной обработке данных.

Методические указания

Очень часто встречается задача заполнить табличную часть какими-нибудь значениями, например, результатом выполнения запроса.

Для заполнения табличной части документа необходимо работать с одноименной таблицей реквизита «Объект», который является основным реквизитом управляемой формы.

Сделаем небольшую учебную задачу: будем заполнять табличную часть документа всей номенклатурой, которая не помечена на удаление. Количество при этом будет равно 1.

Поскольку по условиям задачи, необходимо отобрать всю не помеченную на удаление номенклатуру, то оптимально это сделать при помощи запроса, в котором буду все не помеченные элементы справочника номенклатура. Для этого нужно создать команду, при выполнении которой будет заполняться табличная часть, и разместить её на форме (рис. 11.1).

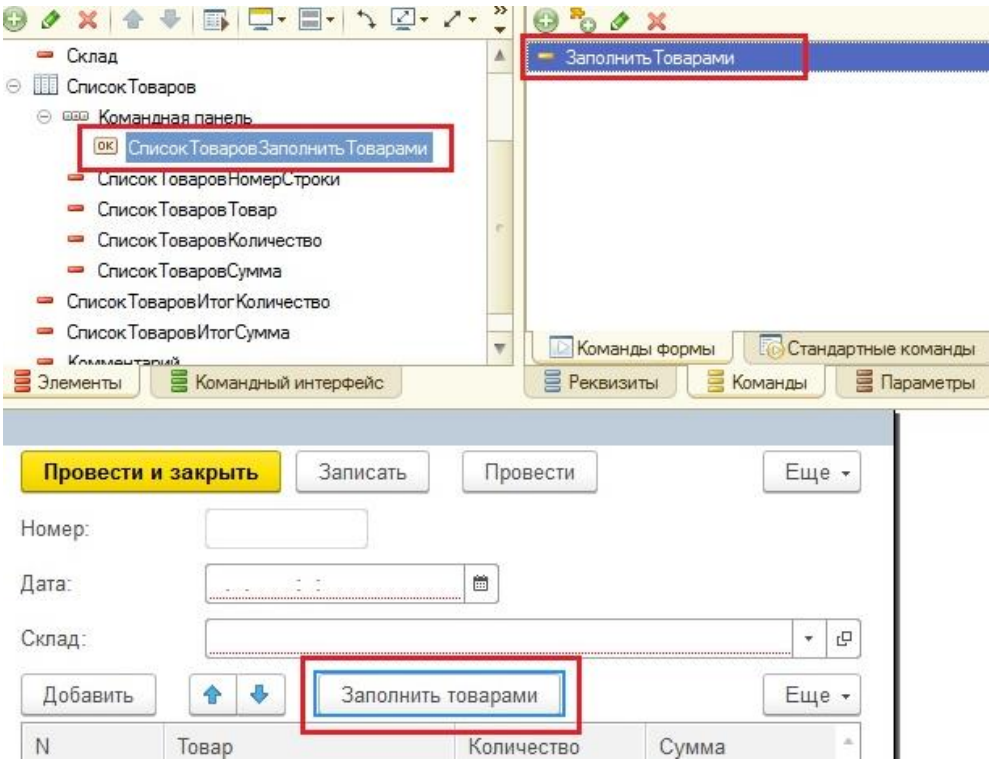

Рисунок 11.1 – Создание команды

Исполнение команды будет выполняться в серверном контексте, поскольку объект Запрос не работает в клиентском контексте (рис. 11.2).

```
&НаСервере
Процедура ЗаполнитьТоварамиНаСервере()

  Запрос = Новый Запрос;
  Запрос.Текст = "ВЫБРАТЬ
          |   Товары.Ссылка КАК Товар,
          |   1 КАК Количество
          |ИЗ
          |   Справочник.Товары КАК Товары
          |ГДЕ
          |   НЕ Товары.ПометкаУдаления";
  Выборка = Запрос.Выполнить().Выбрать();
  Пока Выборка.Следующий() Цикл
    НовСтр = Объект.СписокТоваров.Добавить();
    ЗаполнитьЗначенияСвойств(НовСтр,Выборка);
  КонецЦикла;

КонецПроцедуры

&НаКлиенте
Процедура ЗаполнитьТоварами(Команда)
  ЗаполнитьТоварамиНаСервере();
КонецПроцедуры
```

Рисунок 11.2 – Код для добавления записей

При обходе запроса, используется метод «Добавить()» табличной части объекта. Данный метод является функцией, которая создаёт и возвращает новую строку.

Обратите внимание, поскольку при обходе запроса и заполнения новой строки, используется метод ЗаполнитьЗначенияСвойств, то поля в запросе должны называться так же, как и поля табличной части.

Для отчистки табличной части можно использовать команду: «Объект.СписокТоваров.Очистить()».

В процессе работы может возникнуть необходимость отработать текущую строку табличной части, т. е. ту строку, которая в данный момент выделена.

В этом обработчике код, который будет пересчитывать значения полей текущей строки табличной части (рис. 11.3).

```
&НаКлиенте
Процедура СписокТоваровЦенаПриИзменении(Элемент)
    ТекДанные = Элементы.СписокТоваров.ТекущиеДанные;
    Если ТекДанные = Неопределено Тогда
        Возврат; //если пустая таблица
    КонецЕсли;
    ТекДанные.Сумма = ТекДанные.Количество * ТекДанные.Цена;
КонецПроцедуры
```

Рисунок 11.3 – Код для изменения текущей записи

Часто возникают задачи обойти табличную часть и изменить какие-либо значения в столбцах. В предыдущих примерах мы изменяли текущую строку таблицы, теперь реализуем пример, когда нужно изменить все строки. В этом случае нет необходимости обращаться к элементам формы, можно работать напрямую с основным реквизитом «Объект» (рис. 11.4).

```
&НаКлиенте
Процедура УмножитьВсеНа2(Команда)

  ТабОбх = Объект.СписокТоваров;
  Для Каждого стрТабл из ТабОбх Цикл
    стрТабл.Количество = стрТабл.Количество * 2;
    стрТабл.Сумма = стрТабл.Цена * стрТабл.Количество;
  КонецЦикла;

КонецПроцедуры
```

Рисунок 11.4 – Код для изменения всей табличной части

Варианты заданий

Задание для всех вариантов: добавить в одну из форм созданной ранее базы данных не менее трех различных команд для изменения записей.

ЛАБОРАТОРНАЯ РАБОТА № 12. ОБЪЕКТ ОБРАБОТКИ

Цель лабораторной работы

Получение практических навыков в объекте Обработки.

Методические указания

Обработки 1С позволяют разрабатывать собственные инструменты как для программирования 1С, так и для настройки или управления 1С.

Обработка является инструментом, написанным программистом. Она ничего не умеет делать самостоятельно и не сохраняет данные в базу данных 1С, в отличие, например, от документа, который сохраняется в базу данных и самостоятельно умеет записываться и проводиться, без дополнительного программирования. Функции обработки 1С целиком зависят от программиста, который ее написал.

По функционалу обработки 1С можно поделить на три вида:

– Вспомогательные обработки 1С конфигурации. В каждой типовой конфигурации есть множество обработок. Они используются как дополнительные интерфейсные формы (рабочий стол пользователя, обзор конфигурации), как часть функционала конфигурации (внос начальных остатков, начальное заполнение базы данных, закрытие месяца);

– Объекты (Классы). Это некий класс, имеющий в своем распоряжении набор «переменных» и «функций». Его прелесть состоит в самодостаточности, т. е. в одном классе собрано все, что нужно для выполнения его функций;

– Дополнительные инструменты пользователя и администратора. Существует множество универсальных инструментов для использования обычно администратором базы данных, которые не привязаны к конкретной конфигурации.

Существуют обработки 1С, встроенные в конфигурацию, и внешние обработки 1С.

Встроенные обработки 1С используются программистом, разрабатывающим конфигурацию, ситуативно – то есть они могут быть или выведены в меню пользователя (часть в меню Сервис), или открываться программно из других объектов 1С (например, из формы справочника).

В конфигураторе встроенные в конфигурацию обработки 1С находятся в ветке «Обработки» (рис. 12.1).

Внешние обработки 1С открываются и в конфигураторе, и в «1С:Предприятие» с помощью меню «Файл/Открыть».

Обратите внимание на порядок выполнения модулей. Модуль объекта обработки 1С выполняется автоматически при открытии обработки 1С в режиме Предприятия. Поэтому, если Вы открываете обработку, написанную злоумышленником – она может быть выполнена автоматически без лишних вопросов.

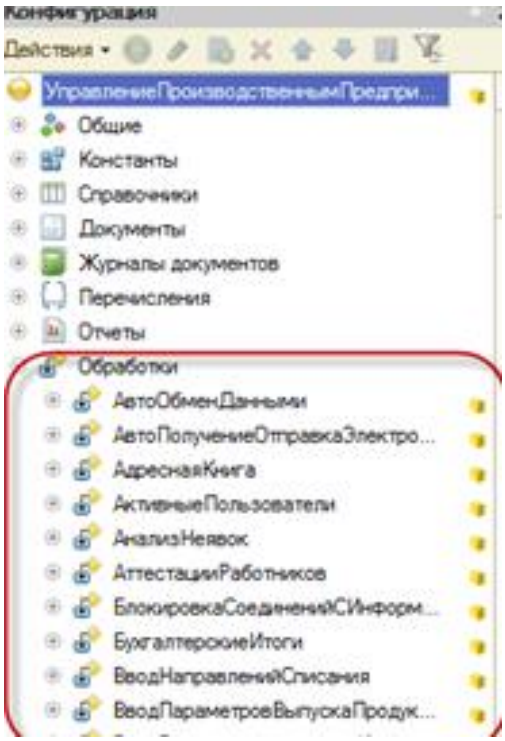

Рисунок 12.1 – Расположение встроенных обработок

Если Вам нужно создать обработку, встроенную в конфигурацию – нажмите правой кнопкой мыши на ветке «Обработки» и выберите «Добавить» (рис. 12.2).

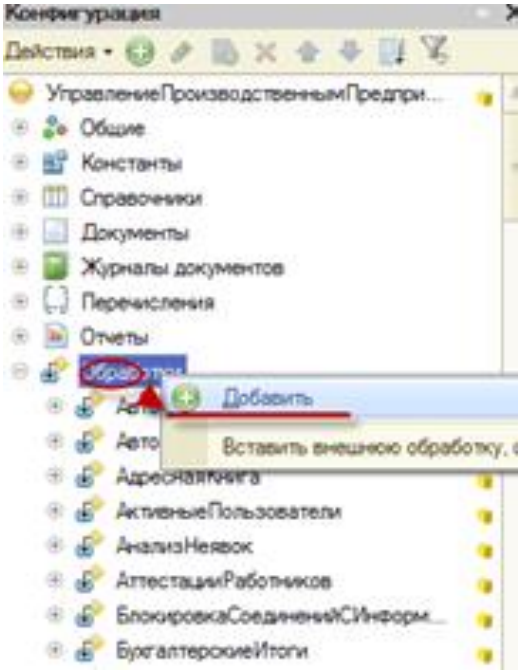

Рисунок 12.2 – Создание встроенной обработки

Если Вам нужно создать внешнюю обработку, выберите «Файл/Новый», в списке вариантов файлов выберите «Обработка» (рис. 12.3).

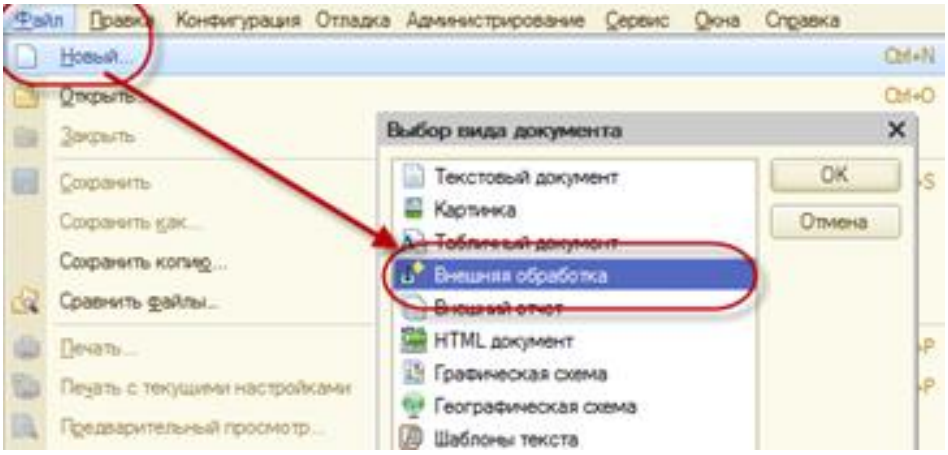

Рисунок 12.3 – Создание внешней обработки

Две основные детали обработки 1С – это [экранная] форма и модуль обработки 1С. В зависимости от предназначения этой конкретной обработки 1С, у нее может не быть или формы (если она используется как класс с функциями) или модуля (если она используется как интерфейсное окно, например рабочий стол пользователя) (рис. 12.4).

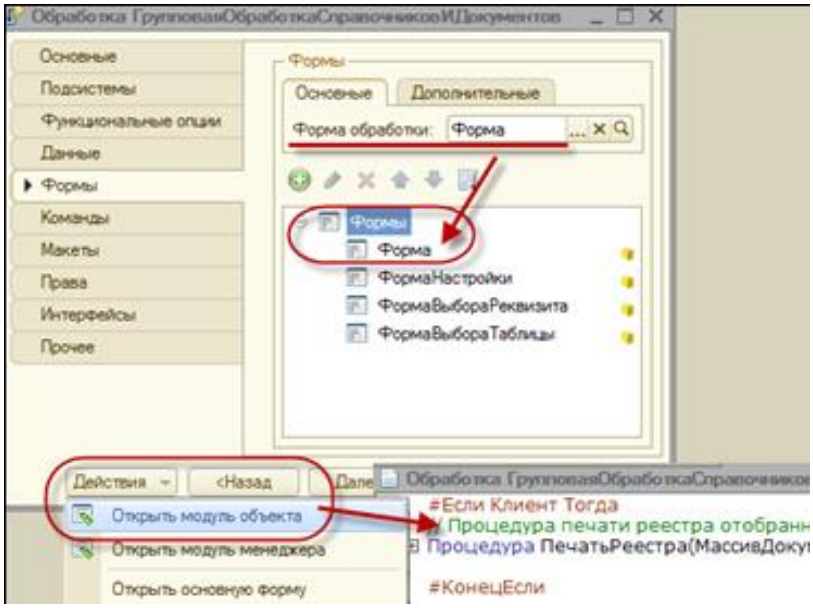

Рисунок 12.4 – Свойства обработки

Публичные функции (которые будут видны извне обработки 1С) должны быть помечены как «Экспорт».

Варианты заданий

Задание для всех вариантов: создать не менее трех обработчиков (один с формой и модулем, один только с формой, один только с модулем) и написать в них код для выполнения внутри созданной ранее базы данных.

ЛАБОРАТОРНАЯ РАБОТА № 13. НАПИСАНИЕ СЛОЖНЫХ ОБРАБОТЧИКОВ СОБЫТИЙ ДЛЯ ДОКУМЕНТОВ. Ч. 1

Цель лабораторной работы

Получение практических навыков в написании сложных обработчиков событий для документов.

Методические указания

Вся необходимая теория была дана в предыдущих лабораторных работах.

Варианты заданий

Задание для всех вариантов: реализовать в созданной ранее базе данных следующие пункты:

1 Создать автоматически подсчитываемые поля: если число в поле больше определенного, то должно выводится текстовое сообщение, если равно ему, то другое и если меньше, то третье.

2 Создать заполнение формы по умолчанию (все поля заполняются любыми символами в соответствии с типом поля) если хоть одно поле не было изменено, вывести об этом сообщение.

3 Вывод определенных колонок на печать.

ЛАБОРАТОРНАЯ РАБОТА № 14. НАПИСАНИЕ СЛОЖНЫХ ОБРАБОТЧИКОВ СОБЫТИЙ ДЛЯ ДОКУМЕНТОВ. Ч. 2

Цель лабораторной работы

Получение практических навыков в написании сложных обработчиков событий для документов.

Методические указания

Вся необходимая теория была дана в предыдущих лабораторных работах.

Варианты заданий

1 Организовать программный подсчет проведенных документов за последние сутки (выводить количество документов и информацию о них).

2 Выгрузка документа в текстовый файл.

ЛАБОРАТОРНАЯ РАБОТА № 15. СОЗДАНИЕ СЛОЖНЫХ ЗАПРОСОВ. Ч. 1

Цель лабораторной работы

Получение практических навыков в написании сложных запросов.

Методические указания

Вся необходимая теория была дана в предыдущих лабораторных работах.

Варианты заданий

Задание для всех вариантов: реализовать в созданной ранее базе данных следующие пункты:

1 Создать запрос с выборкой записей по какому либо параметру (например цена товара от 100 до 1000 рублей), и исключающий другой параметр (например цвет синий).

2 Создать запрос, который выводит записи, числовые значения которых при делении на 3 дадут в результате четное число.

ЛАБОРАТОРНАЯ РАБОТА № 16. СОЗДАНИЕ СЛОЖНЫХ ЗАПРОСОВ. Ч. 2

Цель лабораторной работы

Получение практических навыков в написании сложных запросов.

Методические указания

Вся необходимая теория была дана в предыдущих лабораторных работах.

Варианты заданий

Задание для всех вариантов: реализовать в созданной ранее базе данных следующие пункты:

1 Вывести запись с самым большим количеством символов.

2 Вывести записи даты в полях, которые отличаются от текущей более чем на 2 года.

3 Вывести записи даты, которые соответствуют определенному дню.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Дейнеко, О. В. Программирование в 1С : учебное пособие / О. В. Дейнеко ; ФГБОУ ВО РГУПС. – Ростов н/Д : [б. и.], 2019. – 74 с. – ISBN 978-5-88814-865-5. ЭБС РГУПС (электронный ресурс).

2 Дейнеко, О. В. Программное обеспечение отечественного производства : учебно-методическое пособие для лабораторных и практических работ / О. В. Дейнеко ; ФГБОУ ВО РГУПС. – Ростов н/Д : [б. и.], 2019. – 40 с. ЭБС РГУПС (электронный ресурс).

3 Казак, А. А. Методы вычислений и моделирование в учете и анализе : учебное пособие / А. А. Казак ; РГУПС. – Ростов н/Д : [б. и.], 2009. – 98 с. ЭБС РГУПС (электронный ресурс).

4 Бойко, Э. В. 1С Предприятие 8.0 : универсальный самоучитель / Э. В. Бойко. – Саратов : Ай Пи Эр Медиа, 2010. – 375 с. – ISBN 2227-8397. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: http://www.iprbookshop.ru/957.html.

5 Заика, А. А. Разработка прикладных решений для платформы 1С:Предприятие 8.2 в режиме «Управляемое приложение» : учебное пособие / А. А. Заика. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 238 с. – ISBN 978-5-4497-0925-7. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: http://www.iprbookshop.ru/102061.html.

6 Основы конфигурирования в системе «1С:Предприятие 8.0» : учебное пособие. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 222 с. – ISBN 978-5-4497-0876-2. – Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: http://www.iprbookshop.ru/102027.html.

7 Скороход, С. В. Программирование на платформе 1С: Предприятие 8.3 : учебное пособие / С. В. Скороход. – Ростов-на-Дону : ЮФУ, 2019. – 135 с. – ISBN 978-5-9275-3315-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/141127.

8 Даева, С. Г. Основы разработки корпоративных информационных систем на платформе 1С: Предприятие 8.3 : учебно-методическое пособие / С. Г. Даева. – Москва : РТУ МИРЭА, 2020. – 74 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/163859.

Учебное издание

Ольгейзер Иван Александрович
Игнатьева Олеся Владимировна
Голубенко Евгений Владимирович

ПРОГРАММИРОВАНИЕ В 1С

Печатается в авторской редакции
Технический редактор Т. И. Исаева

Подписано в печать 09.03.2022. Формат 60×84/16.
Усл. печ. л. 3,49. Тираж экз. Изд. № 5011. Заказ .

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

Адрес университета: 344038, г. Ростов н/Д, пл. Ростовского Стрелкового
Полка Народного Ополчения, д. 2, www.rgups.ru