

Лабораторная работа № 4

Разработка информационной системы для выведения информации о кассовых операциях.

1 Задание

Заказчик просит разработать учетную систему для ведения информации о кассовых операциях.

Кассовые операции включают в себя приход и расход денежных средств с обязательным указанием контрагента и суммы.

Необходимо предусмотреть возможность выбирать из списка только те документы, в которых фигурирует выбранный пользователем контрагент.

Кроме того, нужно реализовать возможность просматривать остаток денежных средств в кассе.

Примерный вид журнала кассовых операций:

Дата		Номер	Тип документа	Контрагент	Сумма
01.01.2025		001	Приход	ООО «Мак»	3000
03.01.2025		002	Расход	ООО «Мак»	7000

2 Подготовка

- Создать новую информационную базу (ЛР 4);
- Открыть информационную базу в режиме «Конфигуратор»;
- Открыть окно конфигурации (см. ЛР 2);

3 Ход работы

В первую очередь, следует разобраться с объектами аналитики, которые необходимо хранить в информационной системе. Однозначно нужно хранить информацию о контрагентах – это частные или юридические лица, с которыми будут вестись любые отношения. Для хранения контрагентов использовать справочник.

1. Создать справочник «Контрагенты».
2. Запустить «1С:Предприятие» и добавить несколько контрагентов.
3. Добавить новый документ «ПКО» (Приходный кассовый ордер, или другими словами – поступление денег в кассу).
4. Добавить реквизиты:
 - «Контрагент» (тип – «СправочникСсылка.Контрагенты»). Данный реквизит будет хранить ссылку на элемент справочника «Контрагенты»;
 - «Сумма» (тип - «Число», точность — 2).
5. Запустить конфигурацию в режиме «1С:Предприятие» и посмотреть на изменение в документе.

6. Добавить несколько документов «ПКО».

7. Аналогично документу «ПКО» необходимо создать и документ «РКО» (Расходный кассовый ордер). Для этого выделите нужный документ в окне конфигурации и нажмите на кнопку «Добавить копированием». Данный процесс продемонстрирован на рисунках 1 — 2.

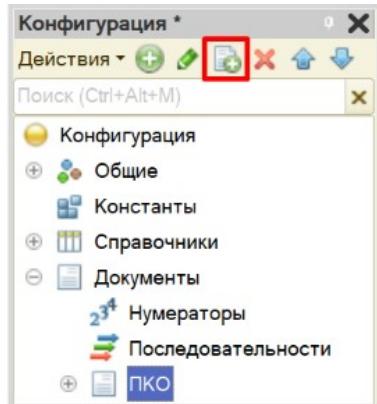


Рисунок 1 — Добавление нового документа путём копирования.

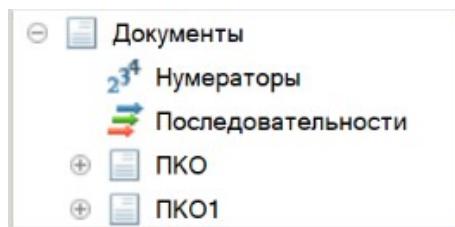


Рисунок 2 - Добавление нового документа путём копирования.

8. Создать несколько документов «РКО».

Таким образом, были реализованы документы, фиксирующие движение денежных средств. Но задача была поставлена так: все документы ПКО и РКО должны находиться в одной таблице. Причем добавляться эти документы должны по мере их создания. Для решения этой задачи необходимо использовать объект конфигурации «Журнал документов».

9. Добавить новый журнал «Кассовые документы». Перейти на вкладку «Данные». Здесь нужно указать, какие документы будут попадать в журнал документов. Чтобы добавить документы требуется нажать на иконку «зелёный карандаш», изображённую на рисунке 3. В открывшемся окне выбрать два созданных ранее документа. Нажать кнопку «OK». Фрагмент формы добавление документов, изображён на рисунке 4.

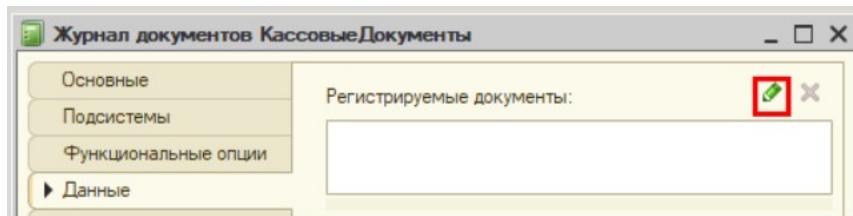


Рисунок 3 — Добавление документов в журнал документов.

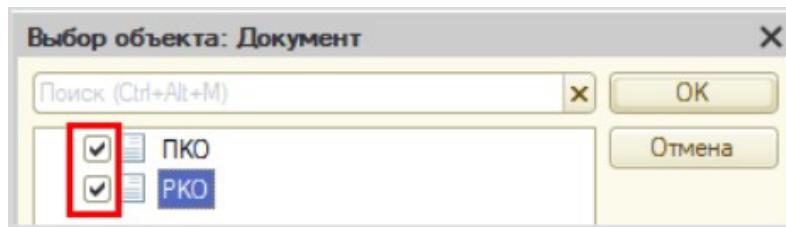


Рисунок 4 — Фрагмент формы добавление документов.

10. Добавить графу «Контрагент» и «Сумма». Указать из каких реквизитов данные будут попадать в графу. Для этого следует нажать на кнопку «многоточие» свойства ссылки, выбрать соответствующий реквизит из обоих документов. Данный процесс изображён на рисунках 5 — 6.

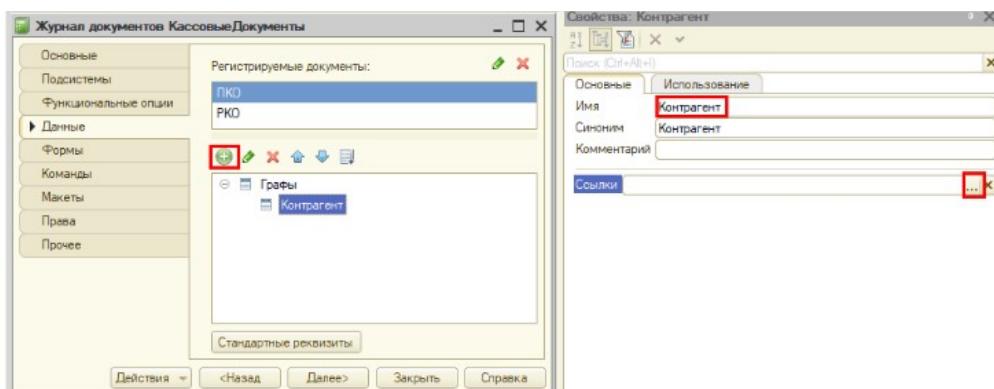


Рисунок 5 — Создание графа.

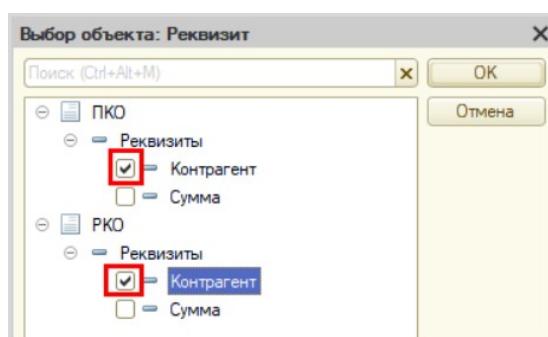


Рисунок 6 — Создание графа.

11. Запустить систему в режиме «1С:Предприятие» и проверить работу журнала.

Можно ли теперь на основе этих документов узнать остаток денег в кассе? Можно, но для этого придется прибегнуть к грубому перебору всех существующих документов. Данный вариант является неправильным, потому что, если таких документов окажется очень много, система будет требовать большого количества ресурсов и времени.

Для решения данной проблемы и ускорения процесса извлечения данных создадим еще один объект – регистр накопления.

12. Создать регистр накоплений и назвать его «Касса», вид регистра выбрать «Остатки». Форма настроек регистра изображена на рисунке 7.

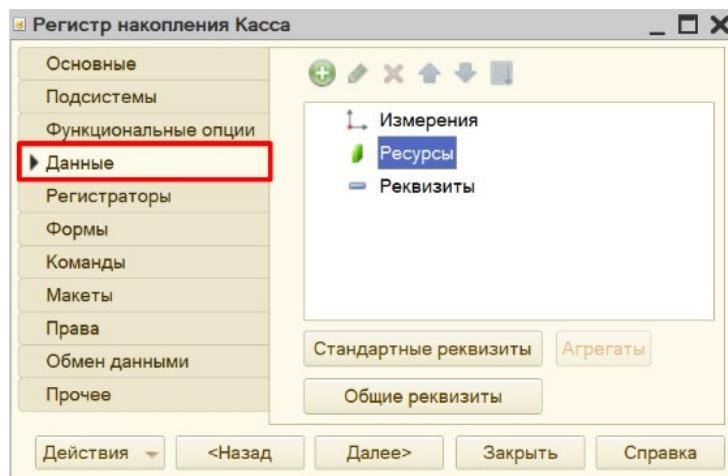


Рисунок 7 — Форма настроек регистра.

13. Добавить ресурс. Чтобы понять, что использовать в качестве ресурса, необходимо задать вопрос: «Что мы хотим накапливать/считать в данном регистре?». Мы хотим считать сумму. Следовательно, сумма и будет являться ресурсом. Тип данного реквизита – «Число». Процесс добавление ресурса изображён на рисунке 8.

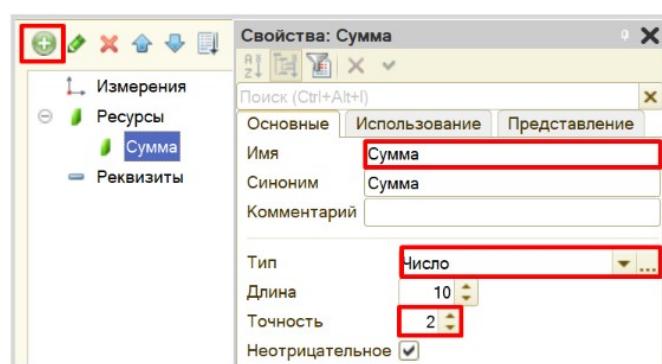


Рисунок 8 — Процесс добавление рисунка.

Чтобы регистр накопления заработал, нужно сделать следующее:

1. Определить источники данных, которые должны попадать в регистр (определить документы-регистраторы).
2. Описать, каким образом данные из документа-регистратора должны попадать в регистр

14. Для определения документов-регистраторов требуется перейти на вкладку «Регистраторы». Здесь нужно выбрать объекты, которые будут передавать данные в регистр. (В нашем случае, это документы «ПКО» и «РКО»).

15. Описать алгоритмы передачи данных для каждого документа. Откройте окно редактирования документа «ПКО» на вкладке «Движения». Воспользуйтесь конструктором движений. Процесс редактирование движения в документе изображён на рисунке 9.

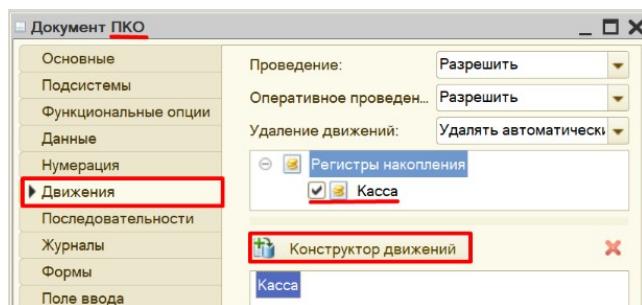


Рисунок 9 — Процесс редактирование движения в документе.

16. Поскольку поступление денежных средств должно увеличивать сумму денег в кассе, то тип движения регистра необходимо выбрать «Приход». Регистр будет обозначаться знаком «+» (плюс). При нажатии на кнопку «Заполнить выражения» реквизиты регистра заполняются автоматически. Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа, которые изображены на рисунке 10.

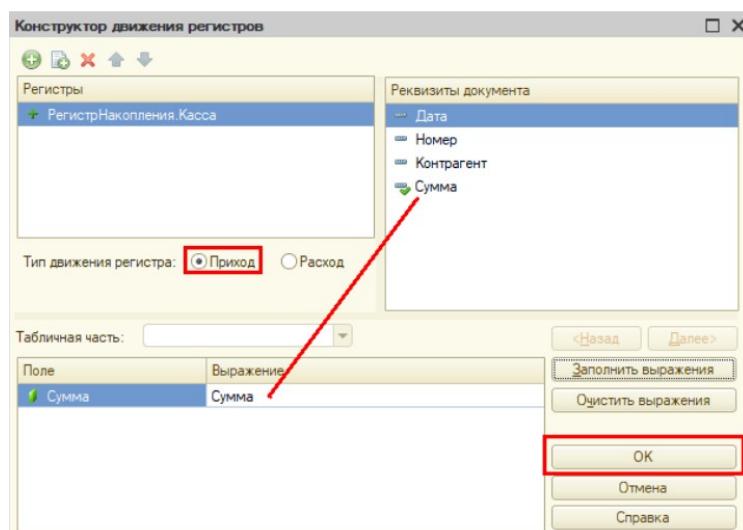


Рисунок 10 — Процесс заполнения конструктора.

При нажатии на кнопку «OK» система сформирует программный код, который при успешном проведении документа сформирует движения в регистр накопления, то есть скопирует данные из документа в регистр накопления.

17. Аналогичным образом создать движения для документа «РКО», при заполнение данных в конструкторе движение, тип движения регистра выбрать как «Расход».

18. Провести заново все документы. В журнале документов выбрать все элементы. Нажать ПКМ и «Провести».

19. Открыть вкладку «Формы» окна редактирования журнала документов и добавьте новую форму журнала документов. Выделив ведущие реквизиты отображения. Процесс создание формы изображён на рисунках 11 и 12.

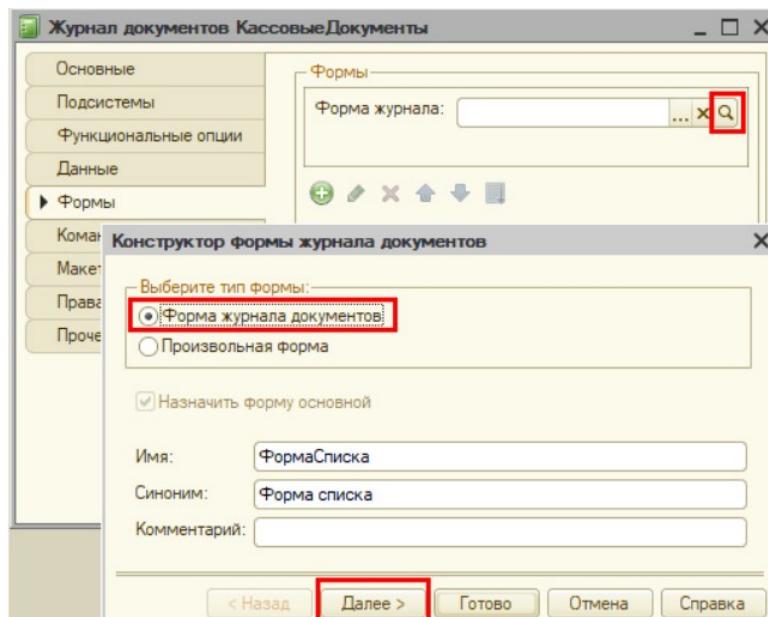


Рисунок 11 — Процесс создание формы журнала документов.

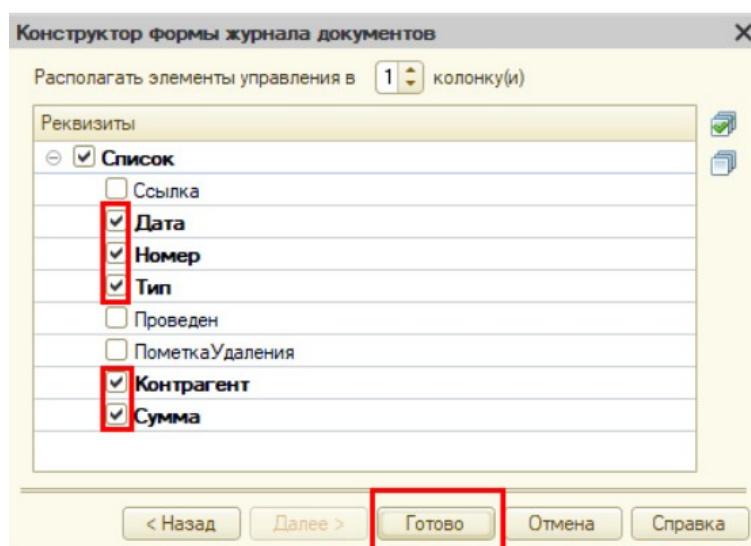


Рисунок 12 — Процесс создание формы журнала документов.

20. Остаток будем получать из регистра накопления «Касса» и выводить его на форму. Чтобы выводить на форму какую-либо информацию, необходимо добавить новый реквизит «Остаток». Форма остатка изображена на рисунке 13.

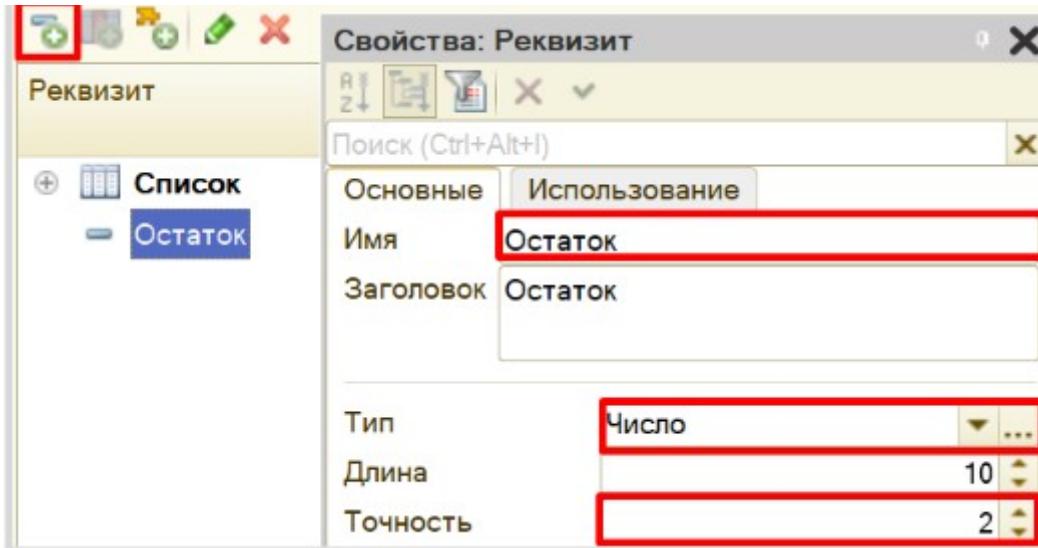
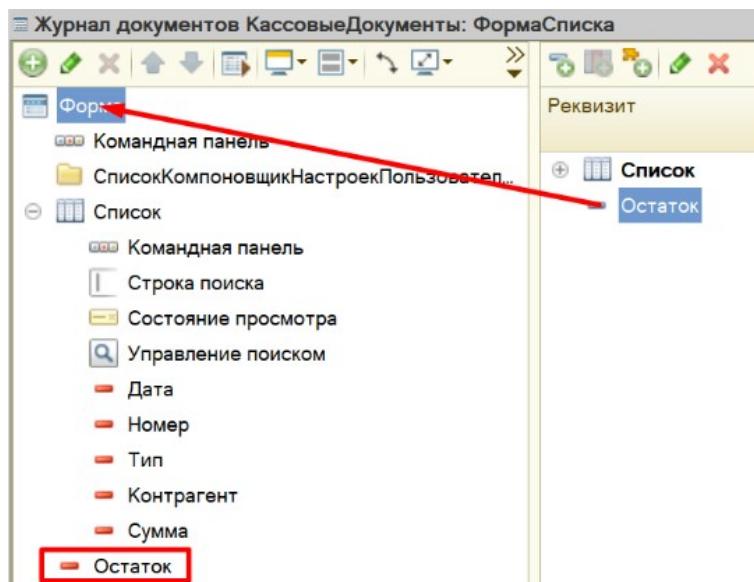


Рисунок 13 — Форма реквизита «Остаток».

21. Чтобы реквизит был виден на форме, удерживая левую кнопку мыши, перенесите его на элемент «Форма», чтобы он расположился под таблицей.



22. Необходимо изменить вид данного элемента. Открываем его свойства и меняем вид с «Поле ввода» на «Поле надписи».

23. Кликните на элемент «Форма» правой кнопкой мыши и в контекстном меню выберем пункт «События». Среди предложенных событий выберем «При Создании На Сервере». Откроется модуль формы, в котором будет представлен шаблон процедуры. Шаблон изображен на рисунке 13.

```
Журнал документов КассовыеДокументы: ФормаСписка

&НаСервере
Процедура ПриСозданииНасервере (Отказ, СтандартнаяОбработка)
    // Вставить содержимое обработчика.
КонецПроцедуры
```

Рисунок 13 — Шаблон процедуры.

24. Вернуться на форму и аналогичным образом для элемента «Список», создать шаблон «ПриИзменение». Для решения поставленной задачи обращаться к серверу нет нужды, поэтому выберем обработчик события на стороне клиента. Модуль формы должен выглядеть как на рисунке 14.

```
Журнал документов КассовыеДокументы: ФормаСписка

&НаСервере
Процедура ПриСозданииНасервере (Отказ, СтандартнаяОбработка)
    // Вставить содержимое обработчика.
КонецПроцедуры

&НаКлиенте
Процедура СписокПриИзменении (Элемент)
    // Вставить содержимое обработчика.
КонецПроцедуры
```

Рисунок 14 — Модуль формы журнала документов.

25. Создать собственную функцию. Для этого спуститесь в самый низ модуля, ниже описанных процедур, набрать «Функ» и нажмите комбинацию клавиш Ctrl+Q. Система предложит вам на выбор несколько вариантов создания шаблона. Нам нужна самая обычная функция. Назвать её «ПолучитьОстатки». Система сформирует шаблон для написания функции. Далее нужно добавить директиву компиляции, то есть дать функции понять, на чьей стороне она будет обрабатываться – на стороне клиента или сервера.

Поскольку данная функция должна будет обращаться к базе данных за получением данных в регистре накопления, а доступ к базе данных всегда имеется только со стороны сервера, то нам нужна директива сервера. Скопируйте ее у процедуры «ПриСозданииНасервере» и поместите перед функцией.

26. Установить курсор внутрь функции. Нажать ПКМ и из контекстного меню выбрать «Конструктор запроса с обработкой результата...». Согласиться на создание нового запроса.

27. На вкладке «Обработка результата» установите тип обработки в значение «Обход результата».

28. Данные будем брать не из регистра накоплений напрямую, а из виртуальной таблицы, которую создает этот регистр автоматически. Эта виртуальная таблица позволит получить уже просуммированные значения по

всем документам. Перенести нужные поля. Ход действий изображён на рисунке 15.

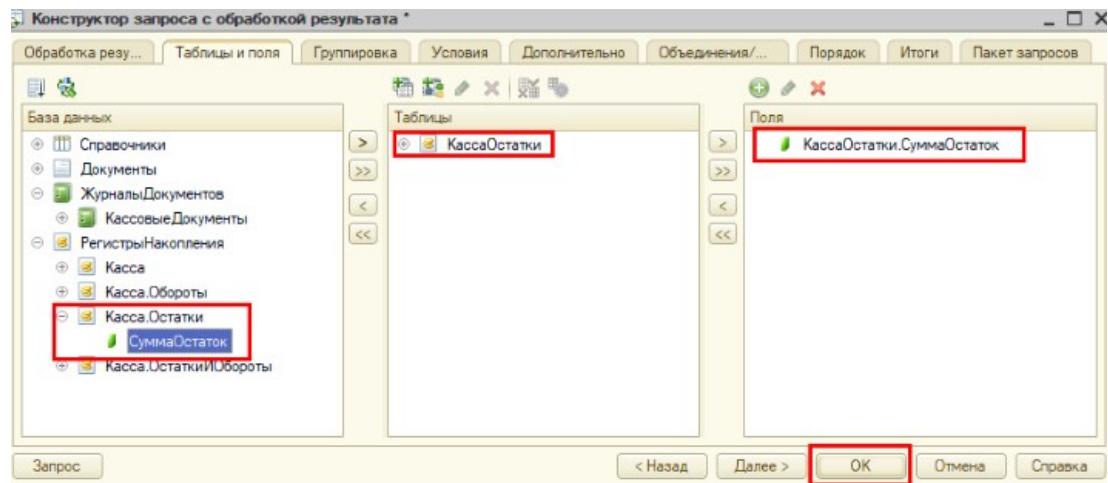


Рисунок 15 — Добавление виртуальной таблицы.

В результате работы с конструктором получен готовый программный модуль, извлекающий сумму остатка денег в кассе.

```
&НаСервере
Функция ПолучитьОстаток()

    //{{КОНСТРУКТОР_ЗАПРОСА_С_ОВРАБОТКОЙ_РЕЗУЛЬТАТА
//  Данный фрагмент построен конструктором.
//  При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

Запрос = Новый Запрос;
Запрос.Текст =
    "ВЫБРАТЬ
    | КассаОстатки.СуммаОстаток КАК СуммаОстаток
    |ИЗ
    | РегистрНакопления.Касса.Остатки КАК КассаОстатки";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    // Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;

//}}КОНСТРУКТОР_ЗАПРОСА_С_ОВРАБОТКОЙ_РЕЗУЛЬТАТА

КонецФункции // ПолучитьОстаток()
```

В результате работы с конструктором получен готовый программный модуль, извлекающий сумму остатка денег в кассе. Запрос будет либо возвращать одно значение, либо ничего не возвращать. Немного изменим логику, которую создал конструктор, и вместо цикла опишем следующее условие:

```

&НаСервере
Функция ПолучитьОстаток()

    //{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | КассаОстатки.СуммаОстаток КАК СуммаОстаток
        |ИЗ
        | РегистрНакопления.Касса.Остатки КАК КассаОстатки";
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

    Если ВыборкаДетальныеЗаписи.Следующий() Тогда
        Возврат ВыборкаДетальныеЗаписи.СуммаОстаток
    Иначе
        Возврат 0
    КонецЕсли;

    //}}КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

Конецфункции // ПолучитьОстаток()

```

29. Функция «ПолучитьОстаток» возвращает значение, которое необходимо передать в созданный ранее реквизит «Остаток». Значение остатка должно быть получено в обеих процедурах – «ПриСозданииНаСервере» и «СписокПриИзменении».

```

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Остаток = ПолучитьОстаток();
КонецПроцедуры

&НаКлиенте
Процедура СписокПриИзменении(Элемент)
    Остаток = ПолучитьОстаток();
КонецПроцедуры

```

30. Проверить работу конфигурации. Вид корректной работы модуля изображён на рисунке 16.

Дата	Номер	Тип документа	Контрагент	Сумма
01.01.2020 12:00:00	00000001	Приходный кассовый ордер	ОАО "Мак"	5 000,00
02.02.2020 15:00:00	00000001	Расходный кассовый ордер	ОАО "Мак"	1 500,00
01.01.2020 13:00:00	00000002	Приходный кассовый ордер	ООО "Василёк"	6 000,00
01.02.2020 14:00:00	00000002	Расходный кассовый ордер	ООО "Василёк"	3 000,00

Остаток: 6 500,00

Рисунок 16 — Работа модуля в конфигурации.

Лабораторная работа № 5

Разработка информационной системы регистрирующей изменение курсов валют.

1 Задание

Требуется разработать информационную систему, регистрирующую изменение курсов валют. В результате выполнения должен получиться график курса валют.



2 Подготовка

- Создать новую информационную базу.
- Открыть информацию в режиме «Конфигуратор».
- Открыть окно конфигурации.

3 Ход работы

1. Создать справочник «Валюты».
2. В окне редактирования справочника на вкладке «Прочие» создать предопределенные элементы: «Доллар», «Франк». Процесс создание предопределённых элементов изображён на рисунках 1 - 3.

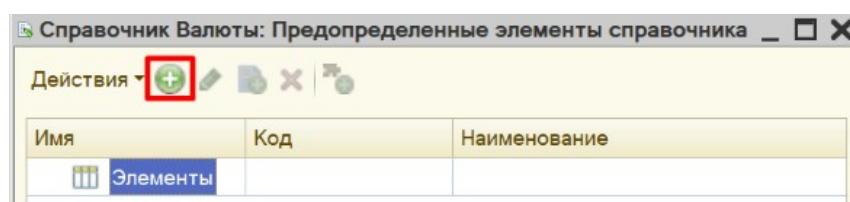


Рисунок 1 — Справочник «Валюты».

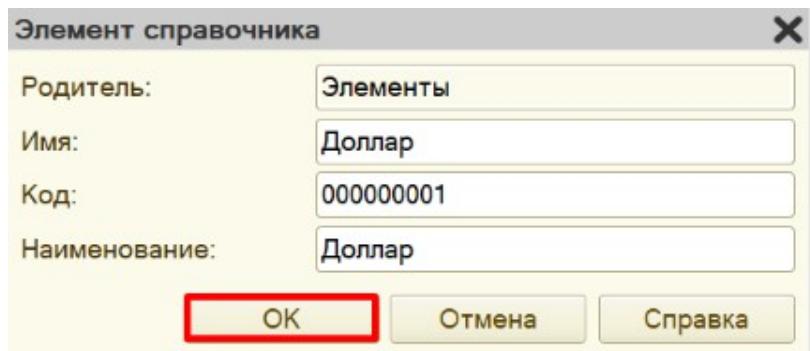


Рисунок 2 — Справочник «Валюты».

Справочник Валюты: Предопределенные элементы справ...		
Действия		
Имя	Код	Наименование
Элементы		
Доллар	000000001	Доллар
Франк	000000002	Франк

Рисунок 3 — Справочник «Валюты».

3. Запустить режим пользователя и убедиться что справочники успешно созданы.

4. Создать валюту «Евро» в справочнике «Валюты».

5. Добавить новый регистр сведений «КурсыВалют». Периодичность регистра установим в значение «В пределах дня». Это значит, что добавлять данные в регистр можно будет только один раз в день. Регистр будет независимым – это означает, что в него записи могут быть добавлены напрямую, без специального документа-регистратора. Форма регистра сведений продемонстрирована на рисунке 4.

Регистр сведений КурсыВалют		
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Основные <input type="checkbox"/> Подсистемы <input type="checkbox"/> Функциональные опции <input type="checkbox"/> Данные <input type="checkbox"/> Регистраторы <input type="checkbox"/> Формы <input type="checkbox"/> Команды <input type="checkbox"/> Макеты 	Имя:	КурсыВалют
	Синоним:	Курсы валют
	Комментарий:	
	Периодичност...	В пределах дня
	Режим записи:	Независимый
	<input checked="" type="checkbox"/> Основной отбор по периоду	
	<input type="checkbox"/>	
	<input type="checkbox"/>	
	<input type="checkbox"/>	

Рисунок 4 — Регистр сведений «КурсыВалют».

6. Открыть вкладку «Данные». Заполнение данного окна всегда проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, следует задать вопрос: «Что нам нужно хранить в данном регистре?». Нам нужно хранить данные о курсах (валют). Следовательно, курс и будет являться ресурсом. Тип данного реквизита – «Число». Основные атрибуты ресурса указаны на рисунке 5.

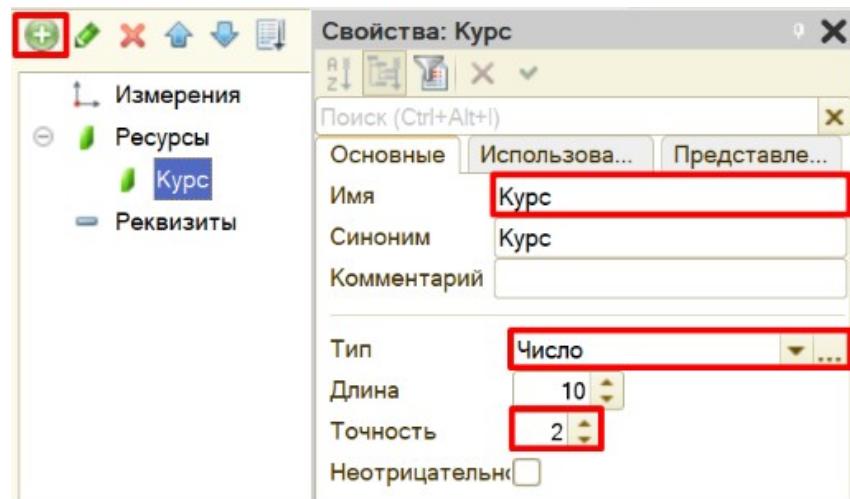


Рисунок 5 — Свойства ресурса «Курс».

Чтобы разобраться с измерением, нужно понять, в разрезе чего мы хотим учитывать курс? Мы хотим хранить данные о курсах (чего?) валют. Значит, в качестве измерения следует добавить реквизит «Валюта». Тип данного реквизита – «СправочникСсылка.Валюты». В этом поле будут храниться ссылки на элементы справочника «Валюты». Основные атрибуты измерения указаны на рисунке 6.

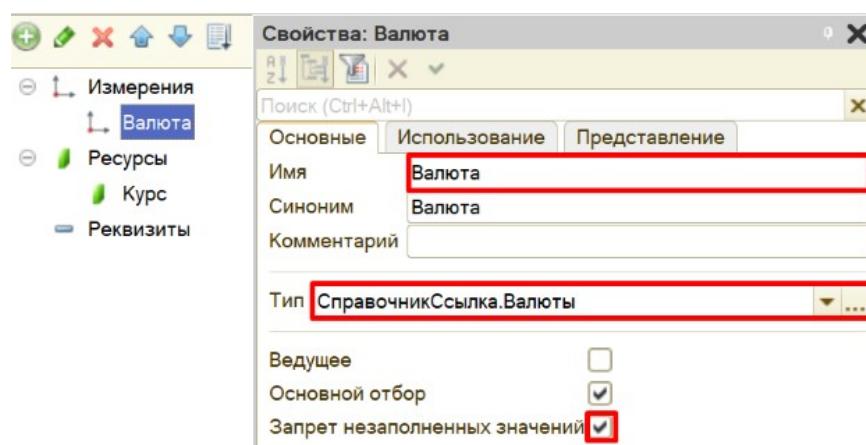


Рисунок 6 — Свойства измерение «Валюта»

7. Открыть программу в режиме «1С:Предприятия» и создать курсы различных валют за несколько дней.

8. Создать новый отчёт «КурсыВалют». Открыть конструктор схемы компоновки данных. Воспользоваться конструктором запросов. Процесс формирования запроса изображён на рисунках 7 — 10.

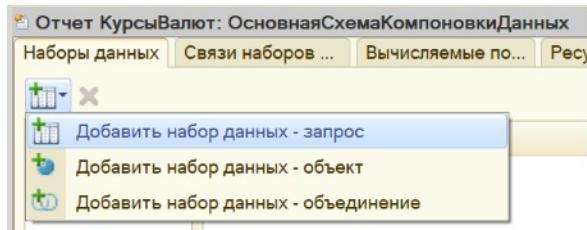


Рисунок 7 — Отчёт «Курсы валют».

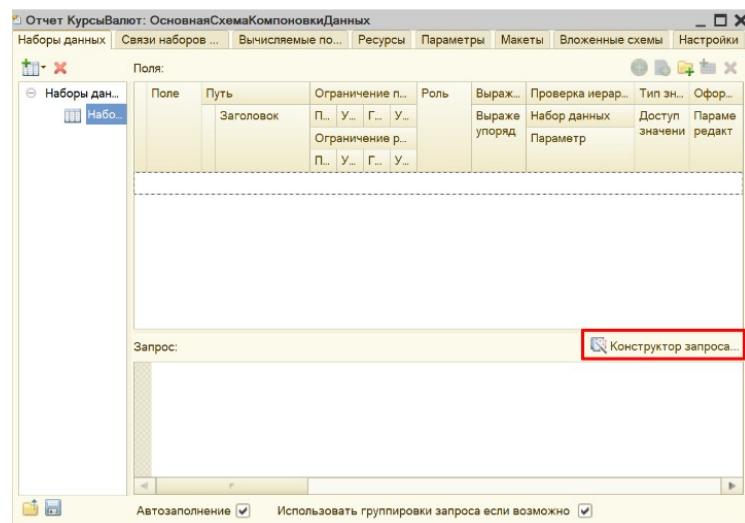


Рисунок 8 — Отчёт «Курсы валют».

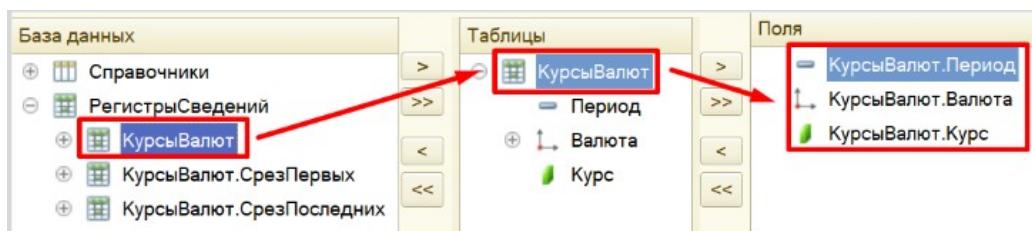


Рисунок 9 - Отчёт «Курсы валют».

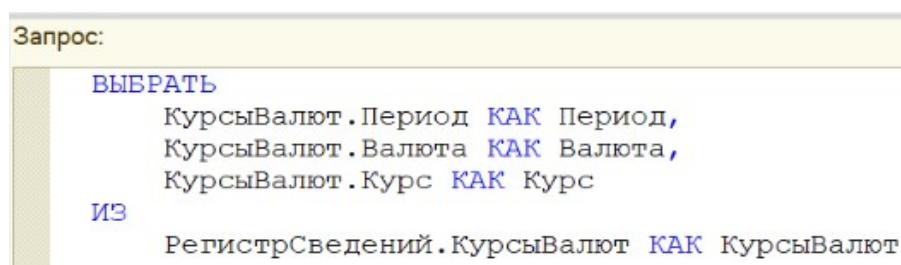
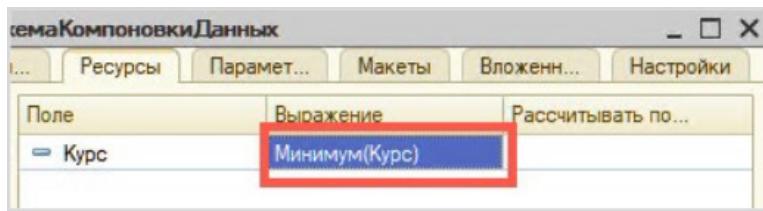


Рисунок 10 - Отчёт «Курсы валют».

9. На вкладке «Ресурсы» определим реквизит «Курс» в качестве ресурса. Вместо выражения по умолчанию установим значение с функцией МИНИМУМ «Минимум(Курс)». Это нужно для корректной работы диаграммы.



10. Перейти в настройки и открыть «Конструктор отчётов». Выбираем тип отчёта «Диаграмма». Выбрать ресурс который будет отображаться на диаграмме. Переходим к следующему этапу настройки диаграммы. Точки и серии – это оси X и Y графика соответственно. Пусть ось Y отображает стоимость валюты на конкретную дату. Даты же пусть будут расположены по оси X. Также нужно отметить поля для упорядочивания. Упорядочим график по периоду, чтобы он строился по возрастанию даты. На последнем шаге выбираем тип — график. Этапы настройки отчёта показаны на рисунках 11 — 15.



Рисунок 11 — Конструктор настроек компоновки данных.

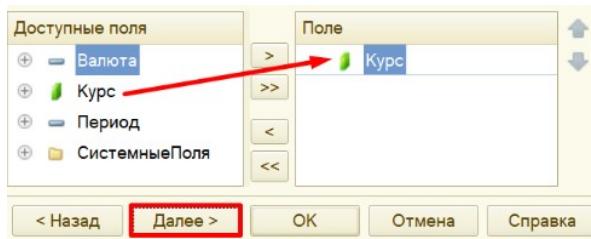


Рисунок 12 — Конструктор настроек компоновки данных.

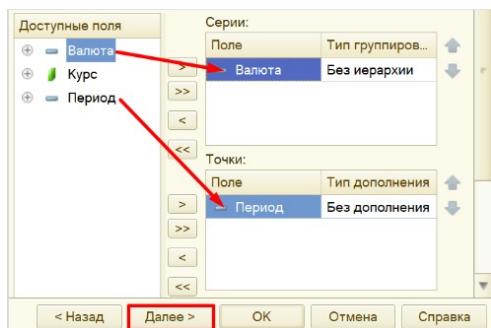


Рисунок 13 — Конструктор настроек компоновки данных.

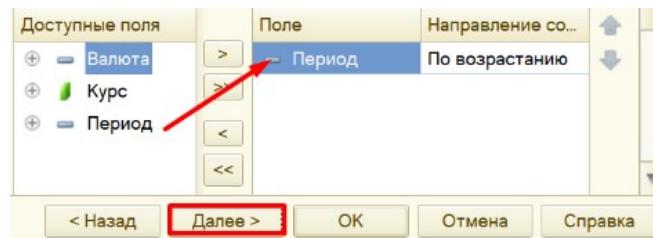


Рисунок 14 — Конструктор настроек компоновки данных.

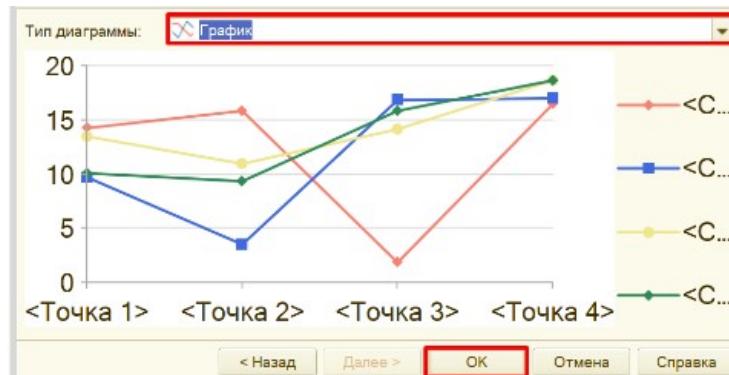


Рисунок 15 — Конструктор настроек компоновки данных.

11. Проверить как выглядит отчёт в «1С: Предприятие»

12. Вернуться в конфигуратор, к настройкам отчета «КурсыВалют». Двойным щелчком мыши откроем настройки точек по периоду. В качестве варианта заполнения укажем «День». Настройка данного процесса продемонстрирована на рисунках 16 и 17.

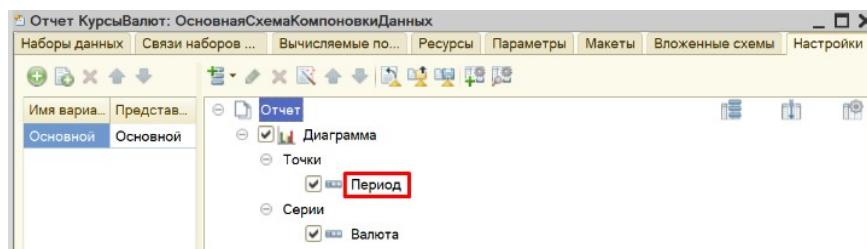


Рисунок 16 — Отчёт «Курсы валют»

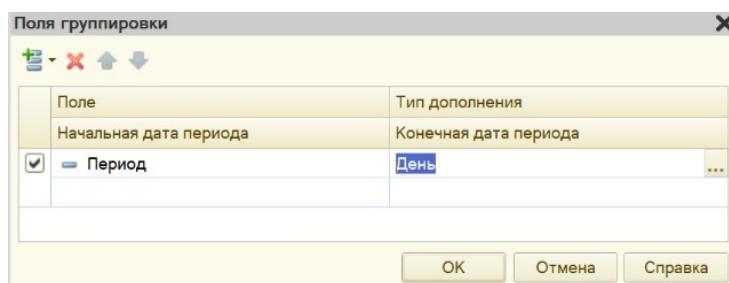


Рисунок 17 — Отчёт «Курсы валют»

13. Проверить как выглядит отчёт в «1С: Предприятие» и выделить отличия

Лабораторная работа № 6

Основы работы с модулем 1С.

1 Задание

Просмотреть основные конструкции в внутреннем языке 1С.

2 Подготовка

- Создать новую информационную базу.
- Открыть информацию в режиме «Конфигуратор».
- Открыть окно конфигурации.

3 Ход выполнения

1. Открыть модуль управляемого приложения из контекстного меню объекта «Конфигурация».

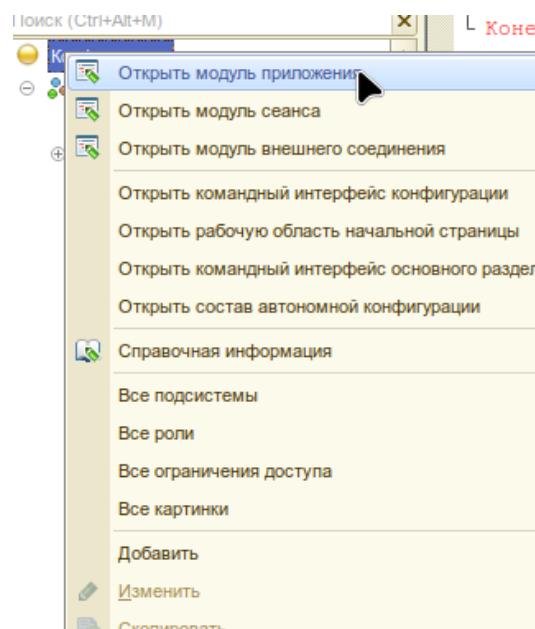


Рисунок 1 — Вид контекстного меню.

2. В открывшемся окне написать следующий код.

```
Сообщить("Привет, мир!");
```

3. Открыть конфигурацию в режиме 1С:Предприятия и просмотреть результат. Пример получившегося результата на рисунке 2.

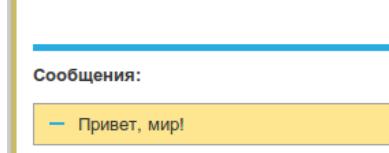


Рисунок 2 — Результат выполнения кода

4. Аналогичным образом вывести на экран стихотворение Александра Сергеевича Пушкина.

Буря мглою небо кроет,
Вихри снежные крутя;
То, как зверь, она завоет,
То заплачет, как дитя.

Обратите внимание, что каждая строка должна выводиться новым сообщением.

5. Основные типы данных используются в языке 1С:

- **Строка.** Отличный пример таких данных - цитата из стихотворения "Буря мглою небо кроет,". Обратите внимание, что данные типа Стока всегда заключаются в **двойные кавычки**.

- **Число.** Числа, могут быть целые и дробные. Пример целого числа: 1000000, дробного - 3.14. Дробная часть от целой отделяется точкой. Числа также имеют знак: плюс или минус.

- **Дата.** Первый полет человека в космос состоялся 12 апреля 1961 года. В языке 1С эти данные будут выглядеть так: '19610412 '. Дата всегда заключается в одинарные кавычки. (К дате можно прибавлять и отнимать секунды)

Требуется ответить на вопросы и вывести эти данные в сообщения пользователя при старте программы:

1. Когда был запущен первый искусственный спутник?
2. Как звали первого космонавта?
3. Сколько планет в солнечной системе?

6. Написать программу которая прибавит к дате 2 марта 2005 года, столько секунд что бы получилось 3 марта 2005 года.

7. Создать переменные с различными типами данных (дата, число, строка) и протестировать их в вызове функции. Пример объявление переменной.

```
КоличествоПланет = 8;
Сообщить("Количество планет: " + КоличествоПланет);
```

8. Требуется написать и вызвать процедуру которая выводит приветствие

```
Процедура НазваниеПроцедуры(Атрибут1, Атрибут2)
    //ТелоПроцедуры
КонецПроцедуры
```

9. Написать функции которые принимает 2 числа и возвращает их сумму.
Вывести результат в сообщениях.

```
Функция НазваниеФункции(Атрибут1, Атрибут2)
    Результат = «»
    Возврат Результат;
КонецФункции
```

10. Написать функцию которая принимает в себя 2 даты и вычисляет сколько дней прошло между этими датами. Протестировать функцию.

11. Создать новый справочник «Товары» с атрибутами цена, вес, описание. Создать форму элемента справочника, открыть его модуль и создать процедуру на Клиенте «ПриОткрытии», написать вывод сообщения, формата

```
НаименованиеТовара весит ВесТовара, и имеет описание ОписаниеТовара
```

Пример создание и код проиллюстрирован на рисунках 3 — 7

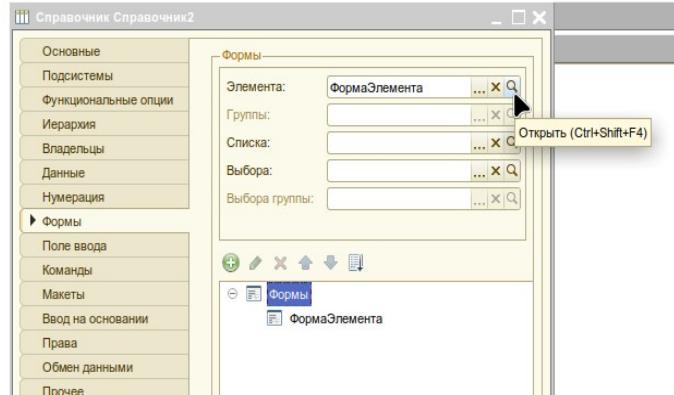


Рисунок 3 — Вид справочника.

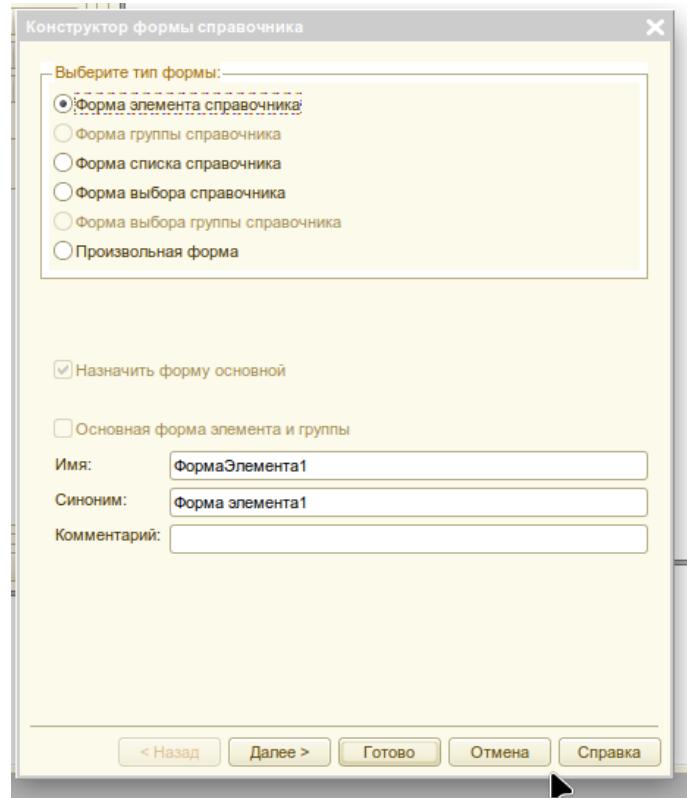


Рисунок 4 — Вид справочника.

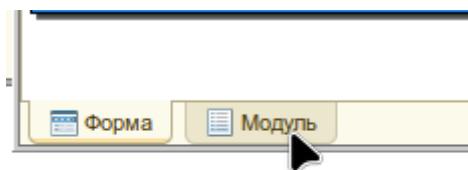


Рисунок 5 — Вид формы справочника.

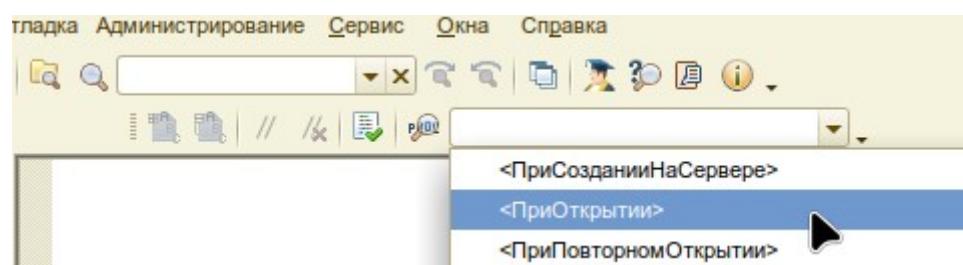


Рисунок 6 — Вид модуля формы справочника.

```
&НаКлиенте
□ Процедура ПриОткрытии(Отказ)
    Сообщить(Объект.Реквизит1);
    КонецПроцедуры
```

Рисунок 7 — Код модуля формы справочника.

Объект в данном случае, является запись которая открывается из справочника, после точке мы можем обратиться к полям данного объекта и манипулировать данными.

12. Создать справочник «Периоды», которые принимает дату начала периода и конца. Создать форму элемента справочника, открыть его модуль и создать процедуру на Клиенте «ПослеЗаписи». Создать на форме новый реквизит с типом число «Разница в днях» и перенести на форму (см ЛР4 п. 21). Перенести функции с шага 10 в данный модуль и выполнить вызов внутри процедуры «ПослеЗаписи» с присвоение к реквизиту «Разница в днях » (см ЛР4 п 29). Протестировать разработанную функциональность.