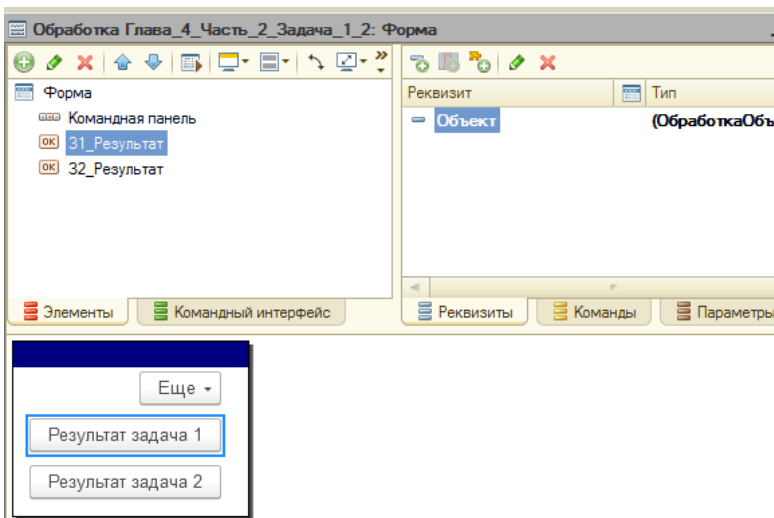


Лабораторная работа № 6

МОДУЛИ

Задание № 1

- Создайте внешнюю обработку,
- В модуле формы обработки создайте две глобальных переменных. Одной присвойте числовое значение, а второй – значение типа **Строка** (вне какого-либо метода).
- Создайте процедуру в модуле формы, в которой эти глобальные переменные будут меняться значениями. Если было: $A = 10$, $B = \text{«Привет»}$, то должно стать:
 $A = \text{«Привет»}$, $B = 10$.
- Создайте команду, в которой будет использоваться процедура, созданная выше,
- В пользовательском режиме проверьте, как будут меняться значения переменных.



```
&НаКлиенте |  
Перем пСтрока, пЧисло;
```

```
&НаКлиенте  
□ Процедура 31_Результат(Команда)  
  Сообщить(пСтрока);  
  31_МеняемМестами();  
  Сообщить(пСтрока);  
КонецПроцедуры
```

```
&НаКлиенте  
□ Процедура 31_МеняемМестами()  
  пСтрока = 1000;  
  пЧисло = "Какая-то строка";  
  Сообщить(пСтрока);  
КонецПроцедуры
```

```
пСтрока = "Какая-то строка";  
пЧисло = 10000;
```

Сообщения:

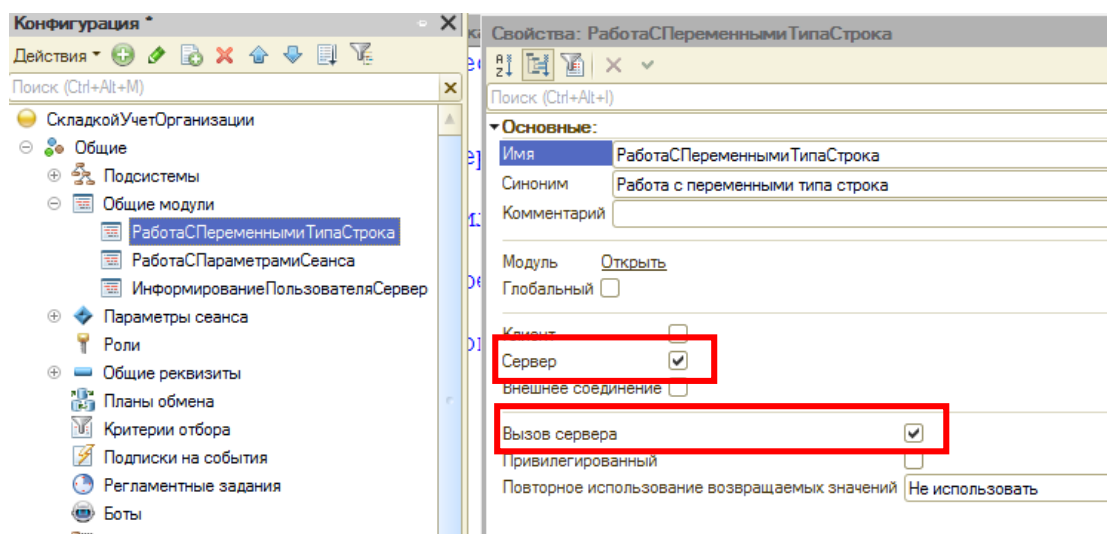
—	Какая-то строка
—	1 000
—	1 000

Задание № 2 *Самостоятельно*

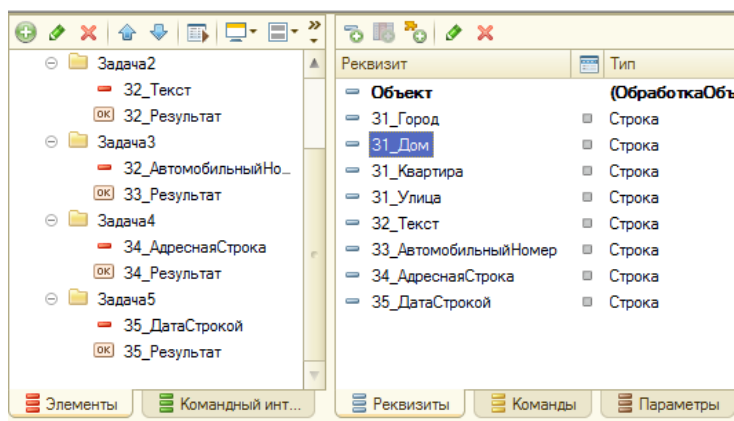
- Создайте внешнюю обработку;
- в модуле формы обработки создайте две глобальных переменных. Одной присвойте числовое значение, а второй – значение типа Строка.
- Создайте процедуру, в которой будут собственные переменные, имеющие такое же название, как и глобальные (можно скопировать код из задания № 1).
- Проверьте, будут ли меняться значения глобальных переменных после выполнения этой процедуры.

Задание № 3

- В конфигурации создайте общий модуль **РаботаСПеременнымиТипаСтрока** (*Выполнение на сервере, Вызов сервера*).



- Перенесите в этот модуль 1-5 задания из лаб. 5.4 (Строковый тип).
- Для каждого задания сделайте отдельные функции (какие параметры должны быть в процедурах, определите самостоятельно, названия функций придумайте самостоятельно), которые будут возвращать значения.
- В этой же конфигурации создайте отдельную обработку (добавьте на панель **Сервис Основного раздела**).
- На форме обработки разместите пять команд, в которых будете использовать созданные из общего модуля (если нужно, создайте реквизиты формы и разместите их на форме).



```

&НаКлиенте
□ Процедура 31_Результат(Команда)
    Сообщить(РаботаСПеременнымиТипаСтрока.ВернутьПолныйАдрес(31_Город,31_Улица,31_Дом,31_Квартира));
КонецПроцедуры

&НаКлиенте
□ Процедура 32_Результат(Команда)
    Сообщить(РаботаСПеременнымиТипаСтрока.ПоменятьТочкиНаПереносСтроки(32_Текст));
КонецПроцедуры

&НаКлиенте
□ Процедура 33_Результат(Команда)
    Сообщить(РаботаСПеременнымиТипаСтрока.РазобратьАвтомобильныйНомер(33_АвтомобильныйНомер));
КонецПроцедуры

&НаКлиенте
□ Процедура 34_Результат(Команда)
    Сообщить(РаботаСПеременнымиТипаСтрока.ВыделитьДомен(34_АдреснаяСтрока));
КонецПроцедуры

&НаКлиенте
□ Процедура 35_Результат(Команда)
    Сообщить(РаботаСПеременнымиТипаСтрока.РазделитьДатуСтрокой(35_ДатаСтрокой));
КонецПроцедуры

□ Функция ВернутьПолныйАдрес(Город, Улица, Дом, Квартира) Экспорт
    Результат = "г. " + Город + ", ул. " + Улица + ", д. " + Дом + ", кв. " + Квартира;
    Возврат Результат;
КонецФункции

□ Функция ПоменятьТочкиНаПереносСтроки(Текст) Экспорт
    Результат = СтрЗаменить(Текст, ".", " ", Символы.ПС);
    Возврат Результат;
КонецФункции

□ Функция РазобратьАвтомобильныйНомер(АвтомобильныйНомер) Экспорт
    ПервыеБуквы = Сред(АвтомобильныйНомер, 1, 2);
    Цифры = Сред(АвтомобильныйНомер, 3, 3);
    ПоследняяБуква = Сред(АвтомобильныйНомер, 6, 1);
    КодРегиона = Сред(АвтомобильныйНомер, 9, 2);

    Результат = "номер " + АвтомобильныйНомер + " состоит
    | первые буквы - " + ПервыеБуквы + "
    | цифры - " + Цифры + "
    | последняя буква - " + ПоследняяБуква + "
    | код региона - " + КодРегиона + ".";

    Возврат Результат;
КонецФункции

□ Функция ВыделитьДомен(АдреснаяСтрока) Экспорт

    НомерВхожденияДвухЧерт = СтрНайти(АдреснаяСтрока, "//");
    СтрокаБезПротокола = Сред(АдреснаяСтрока, НомерВхожденияДвухЧерт + 2, СтрДлина(АдреснаяСтрока));
    НомерСледующейЧерты = СтрНайти(СтрокаБезПротокола, "/");
    ДоменПредварительный = Сред(СтрокаБезПротокола, 1, НомерСледующейЧерты - 1);
    ДоменБезWWW = СтрЗаменить(ДоменПредварительный, "www.", "");

    Возврат ДоменБезWWW;

КонецФункции

□ Функция РазделитьДатуСтрокой(ДатаСтрокой) Экспорт

    День = Сред(ДатаСтрокой, 1, 2);
    Месяц = Сред(ДатаСтрокой, 4, 2);
    Год = Сред(ДатаСтрокой, 7, 4);
    Результат = "день - " + День + "
    | месяц - " + Месяц + "
    | год - " + Год;

    Возврат Результат;

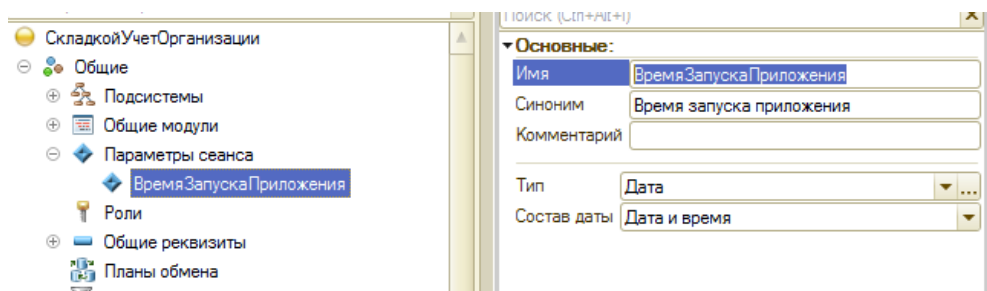
КонецФункции

```

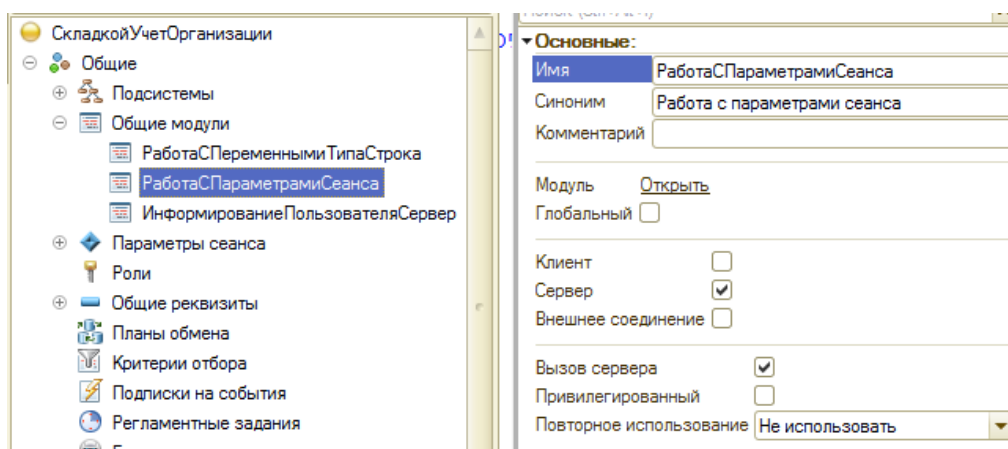
Задание № 4.

В конфигурации реализуйте хранение времени начала запуска приложения и способ чтения:

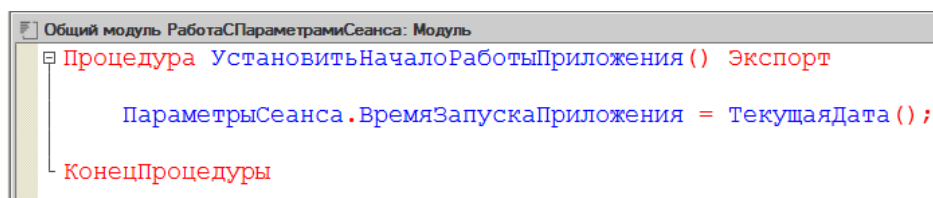
- а) Создайте параметр сеанса **ВремяЗапускаПриложения** (тип **Дата**, состав даты – **Дата и Время**).



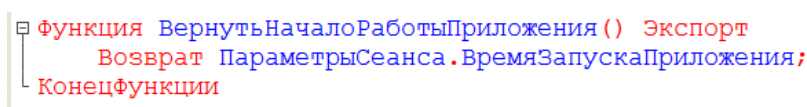
- б) Создайте общий модуль **РаботаСПараметрамиСеанса** (сервер, вызов сервера).



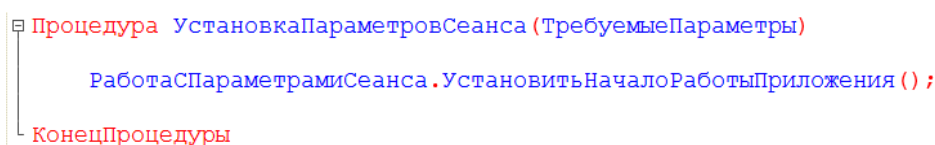
- в) В общем модуле создайте экспортную процедуру, в которой будете устанавливать значение текущей даты параметру сеанса.



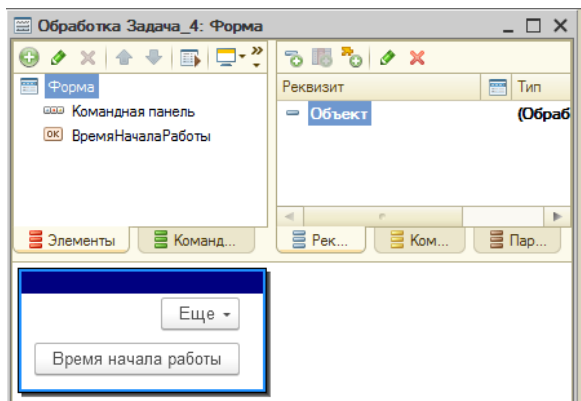
- г) Создайте экспортную функцию, которая будет возвращать значение параметра сеанса.



- е) В модуле сеанса в обработчике **УстановкаПараметровСеанса** вызывайте процедуру из общего модуля, которая устанавливает значение параметру сеанса.



- ф) Создайте обработку, в которой будете получать время запуска приложения при помощи соответствующей функции из общего модуля. Выводите только время запуска в окне сообщений.



```

&НаКлиенте
Процедура ВремяНачалаРаботы(Команда)
    Сообщить(Формат(РаботаСПараметрамиСеанса.ВернутьНачалоРаботыПриложения(), "ДЛФ=T"));
КонецПроцедуры

```

Задача № 5.

В модуле приложения в обработчике `ПриНачалеРаботыСистемы()` проверяйте текущее время, и если оно не попадает в интервал от 8 до 17 часов, то выводите сообщение, что рабочий день ещё не начался.

```

Процедура ПриНачалеРаботыСистемы()
    ТекущееВремя = Дата(1,1,1,Час(ТекущаяДата()),Минута(ТекущаяДата()),Секунда(ТекущаяДата()));
    Если ТекущееВремя < Дата(1,1,1,8,0,0) или ТекущееВремя > Дата(1,1,1,17,0,0) Тогда
        Сообщить("Рабочий день или не начался, или уже прошёл");
    КонецЕсли;
КонецПроцедуры

```

Задача № 6.

- В модуле менеджера справочника `Номенклатура` создайте экспортную процедуру `ИнформированиеПриЗаписи`, в которой просто выводите какое-то сообщение, что номенклатура записана успешно.
- В модуле объекта справочника `Номенклатура` создайте обработчик события `ПриЗаписи`, в котором и обращайтесь к процедуре менеджера справочника.

```

Справочник Номенклатура: Модуль менеджера
Процедура ИнформированиеПриЗаписи() Экспорт
    Сообщить("Номенклатура записана");
КонецПроцедуры

```

```

Справочник Номенклатура: Модуль объекта
Процедура ПриЗаписи(Отказ)
    Справочники.Номенклатура.ИнформированиеПриЗаписи();
КонецПроцедуры

```

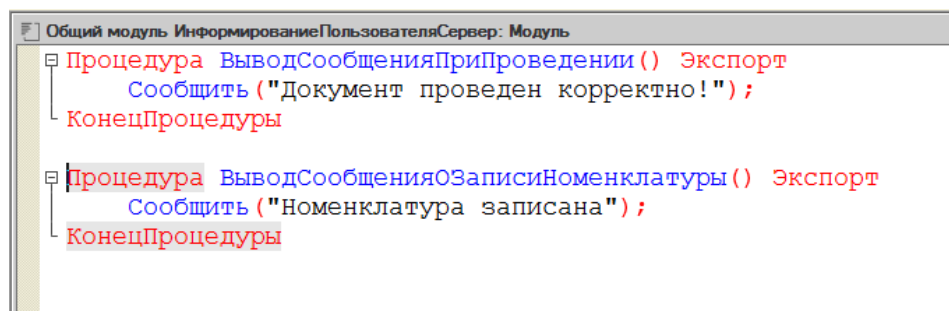
Задача № 7. *Самостоятельно*

- В документах **ПриходноКассовыйОрдер** и **РасходноКассовыйОрдер** были созданы обработчики проведения документов.
- Создайте для этих документов систему информирования:
 - Создайте серверный общий модуль **ИнформированиеПользователяСервер**,
 - В этом модуле создайте процедуру **ВыводСообщенияПриПроведении**, в которой будет сообщаться, что всё проведено корректно.
 - В обработчиках **ПриПроведении** документов **ПриходноКассовыйОрдер** и **РасходноКассовыйОрдер** необходимо вызывать эту процедуру.
 - Проверьте, как работает проведение документов.

Задача № 8.

Переделайте задачу 6:

- создайте процедуру о записи номенклатуры в общем модуле из задачи 7,
- и уже эту процедуру используйте в методе менеджера объекта.



Вызов процедуры в модуле менеджера объекта осуществить самостоятельно!