

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский Томский государственный  
университет»**

**«1С программист»**

Самостоятельная работа №1  
Темы "Переменные и операторы", "Процедуры и функции",  
"Примитивные типы данных".  
"1С программист"

Выполнила : ст.гр. 1С-808

Кононова Виктория Юрьевна

2022

## Введение

Темы "Переменные и операторы", "Процедуры и функции", "Примитивные типы данных" - это первые темы в курсе обучения 1С программистов. Цель самостоятельной работы по данным темам: закрепление знаний о типах данных "Строка", "Число", "Дата", "Булево", об условных операторах, о циклах, о функциях и процедурах. В рамках самостоятельной работы на базе учебной платформы "1С:Предприятие" была создана конфигурация (простейшее прикладное решение) с обработками, выполняющими отдельные задачи из задания. Каждой задаче соответствует своя обработка. Код решения задач находится в модулях форм обработок. Данный отчет предназначен для пояснения решения, содержит скриншоты и код.

### 1 Текст задания

1. Создать функцию, которая выводит факториал заданного числа. Факториал должен вычисляться с помощью рекурсии. Должна быть предусмотрена проверка вычисления факториала с помощью цикла.

2. Проинициализировать три переменные: а) СтудентФИО- записать в переменную строку с полным ФИО студента (например: "Петров Пётр Петрович"); б) ДатаРожденияСтудента - записать дату рождения; в) ВозрастСтудента - число полных лет. Вывести на экран сообщение в следующем формате : "Студент Петров П.П. 1980 г.р. 42 года".

3. Создать функцию, которая возводит число в степень. В функции два аргумента. Первый аргумент - число, которое нужно возвести в степень. Второй аргумент - степень числа. Реализовать через умножение. Использовать функцию возведения в степень для проверки.

4. Написать функцию, которая в зависимости от типа переданного параметра выводит сообщение.

Если типы "число", "строка", "дата", то сообщение: "Данный параметр имеет тип " и далее название типа.

Если тип "булево", то сообщение: "Данный параметр может принимать значение "истина" или "ложь".

Если другой тип, то сообщение: "тип не определен".

5. Самостоятельно ознакомиться с конструкцией Попытка-Исключение-КонецПопытки. Привести примеры использования.

6\*. Ознакомиться с определениями "Функция", "Процедура" в синтаксис-помощнике.

### 2 Описание решения

#### 2.1 Начальная страница

Для решения создана новая база с новой конфигурацией. Создана обработка "Приветствие". Эта обработка использована как рабочая область начальной страницы. Содержит "Надписи-декорации" с информацией о задании. Вид рабочей области начальной страницы в интерфейсе пользователя представлен на рисунке 1.

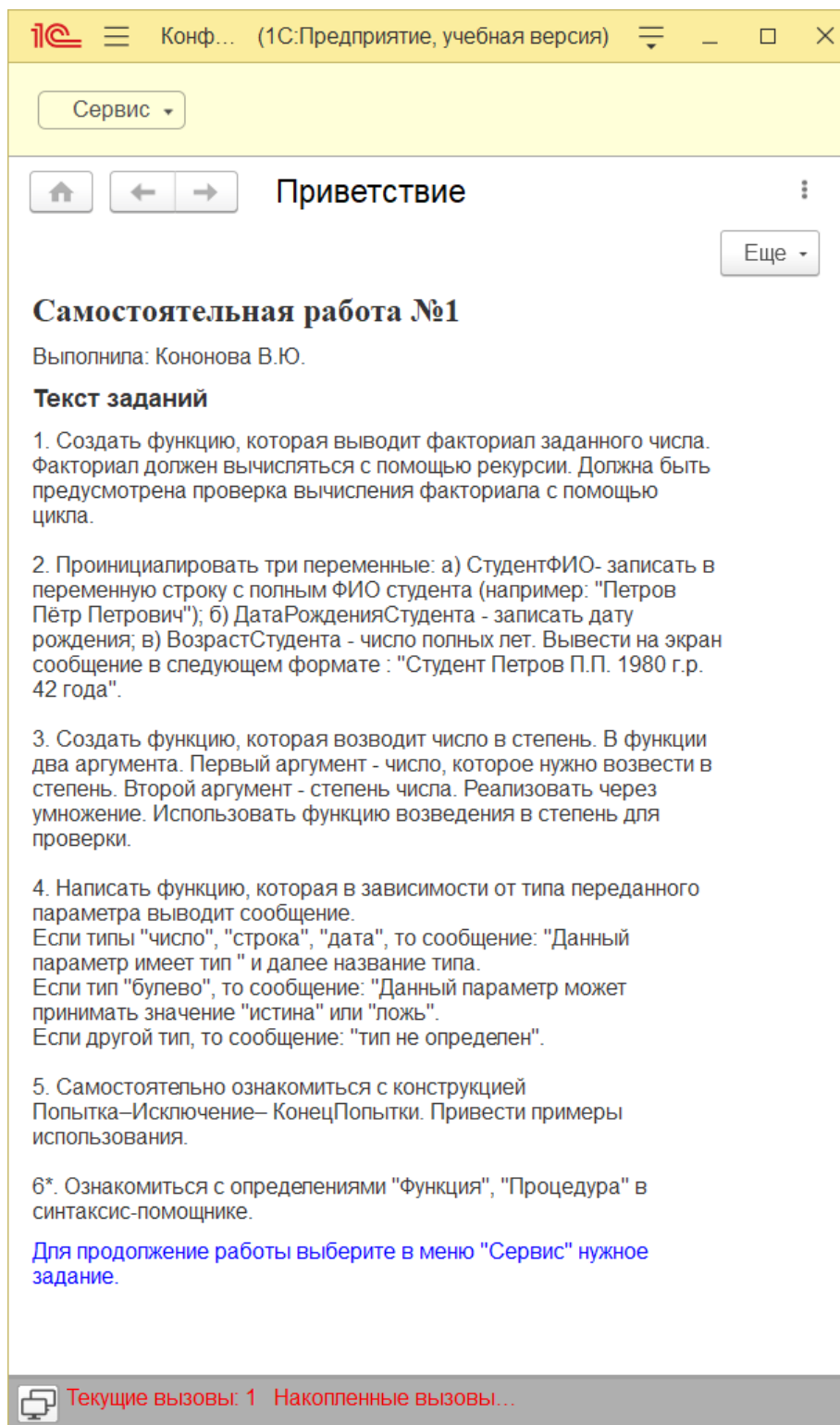


Рисунок 1 - Обработка "Приветствие"

Для удобства каждая из шести задач выполнена в отдельной обработке (рисунок 2).

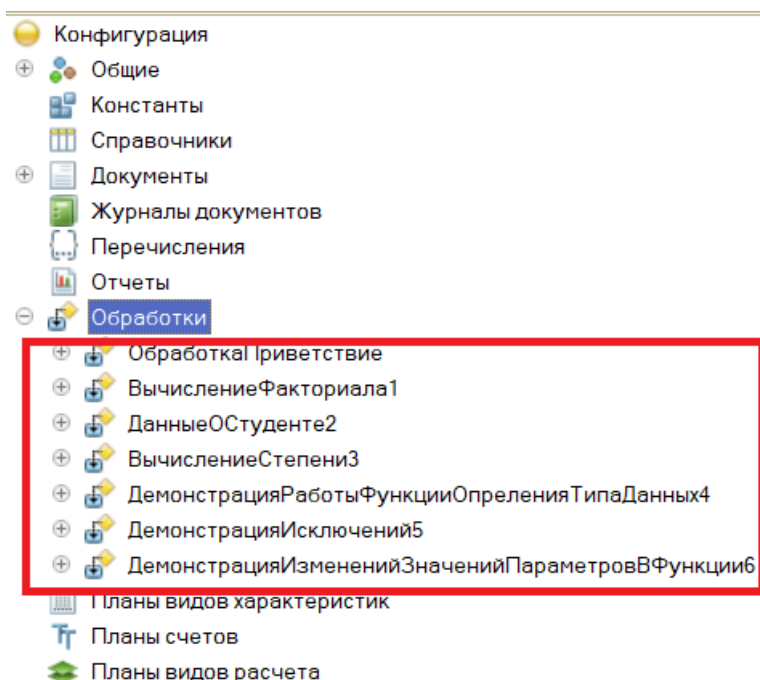


Рисунок 2 - Обработки в дереве конфигурации

В интерфейсе пользователя к каждому заданию можно перейти, выбрав в меню "Сервис" (рисунок 3) соответствующий пункт.

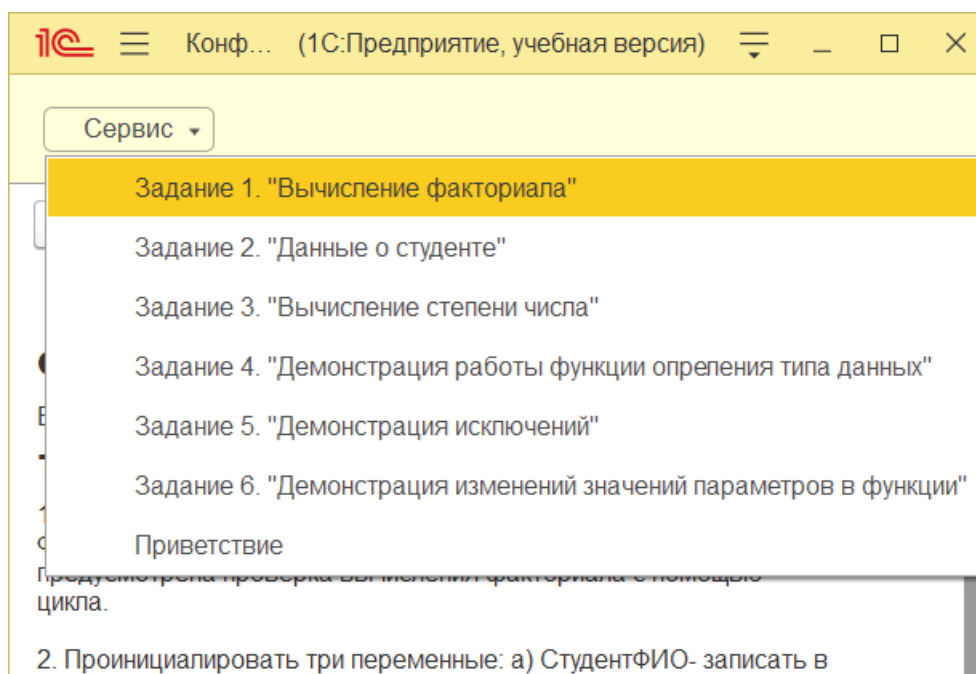


Рисунок 3 - Пункты меню "Сервис"

## 2.2 Вычисление факториала

Выбрав в меню "Сервис" Задание 1 переходим в обработку "Вычисление факториала" (рисунок 4). В поле для ввода необходимо ввести натуральное число и нажать кнопку "Вычислить факториал".

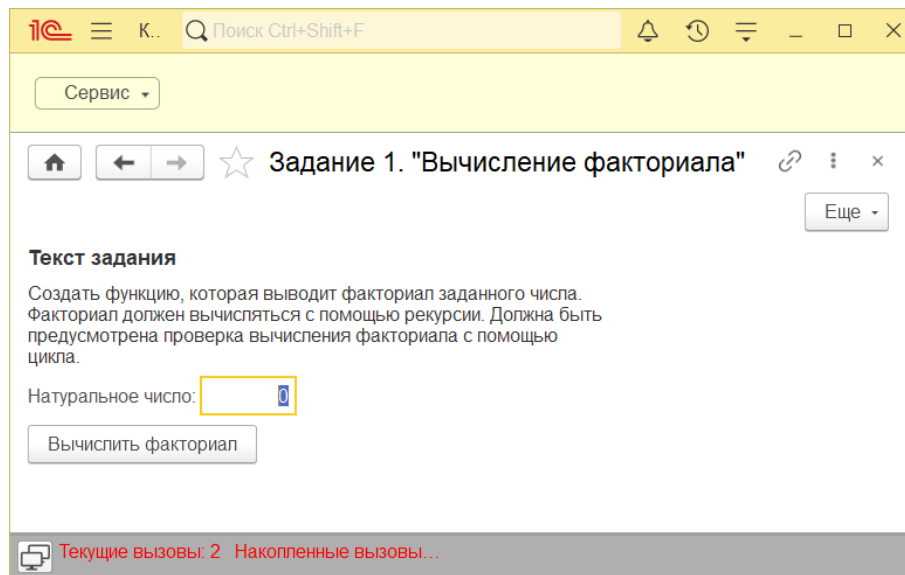


Рисунок 4 - Обработка "Вычисление факториала"

Пример расчета факториала для числа 5 представлен на рисунке 5.

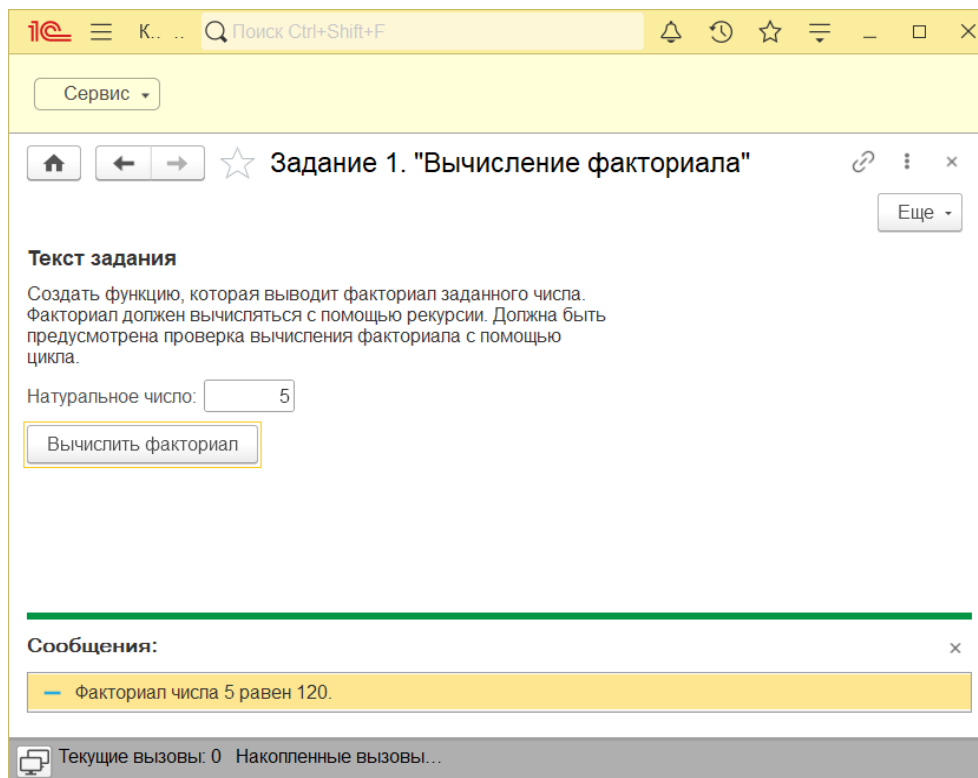


Рисунок 5 - Пример вычисления факториала

На форме обработки был создан реквизит "НатуральноеЧисло" и команда "ВычислитьФакториал" (рисунок 6).

Обработка ВычислениеФакториала1: Форма

Форма

- Командная панель
- Декорация1
- Декорация2
- НатуральноеЧисло
- ВычислитьФакториал

Реквизит

Объект	Тип
НатуральноеЧисло	Число

Текст задания

Создать функцию, которая выводит факториал заданного числа. Факториал должен вычисляться с помощью рекурсии. Должна быть предусмотрена проверка вычисления факториала с помощью цикла.

Натуральное число:

Вычислить факториал

Рисунок 6 - Форма обработки

Модуль формы обработки содержит код решения задачи (рисунок 7).

Обработка ВычислениеФакториала1: Форма

```
&НаКлиенте
[ Процедура ВычислитьФакториал (Команда) ] ...

&НаКлиенте
[ Процедура ДелатьВычисления (НатурЧисло) ] ...

&НаКлиенте
[ Функция ФакториалЦикл (Значение) ] ...

&НаКлиенте
[ Функция Факториал (Значение) ] ...

&НаКлиенте
[ Процедура ВыводОтвета (Число, Факториал) ] ...

&НаКлиенте
[ Процедура ВыводОшибки (Число, Факториал, ФакториалЦикл) ] ...
```

Рисунок 7 - Модуль формы обработки с кодом решения задачи

Код решения представлен ниже (немного съезжает текст из-за ширины страницы А4 отчета).

&НаКлиенте

Процедура ВычислитьФакториал(Команда)

НатурЧисло = НатуральноеЧисло;

Если НатурЧисло < 1 Тогда  
Сообщить("Факториал рассчитывается только для целых чисел, которые больше 0.");

ИначеЕсли НатурЧисло = 1 Тогда  
Сообщить("Факториал числа 1 равен 1.");

Иначе  
ДелатьВычисления(НатурЧисло);  
КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ДелатьВычисления(НатурЧисло)

ИскомыйФакториал1 = Факториал(НатурЧисло);  
ИскомыйФакториал2 = ФакториалЦикл(НатурЧисло);

Если ИскомыйФакториал1 = ИскомыйФакториал2 Тогда  
ВыводОтвета(НатурЧисло, ИскомыйФакториал1);  
Иначе

ВыводОшибки(НатурЧисло, ИскомыйФакториал1, ИскомыйФакториал2);  
КонецЕсли;

КонецПроцедуры

&НаКлиенте

Функция ФакториалЦикл(Значение)

Результат = 1;

Для Индекс = 1 По Значение Цикл  
Результат = Результат \* Индекс;  
КонецЦикла;

Возврат Результат;

КонецФункции

&НаКлиенте

Функция Факториал(Значение)

Если Значение = 1 Тогда  
Возврат 1;  
Иначе

```
        Возврат Значение * Факториал(Значение - 1)
    КонечЕсли;
```

```
КонечФункции
```

```
&НаКлиенте
```

```
Процедура ВыводОтвета(Число,Факториал)
```

```
    Сообщить("Факториал числа " + Число + " равен " + Факториал + ".");
```

```
КонечПроцедуры
```

```
&НаКлиенте
```

```
Процедура ВыводОшибки(Число,Факториал,ФакториалЦикл)
```

```
    Сообщить("Ошибка в вычислениях. Факториал числа " + Число + ", вычисленный рекурсивно, равен " + Факториал + ". А факториал, вычисленный в цикле равен " + ФакториалЦикл + ".");
```

```
КонечПроцедуры
```

Текст рекурсивной функции представлен ниже.

```
&НаКлиенте
```

```
Функция Факториал(Значение)
```

```
    Если Значение = 1 Тогда
```

```
        Возврат 1;
```

```
    Иначе
```

```
        Возврат Значение * Факториал(Значение - 1)
```

```
    КонечЕсли;
```

```
КонечФункции
```

Проверка решения с помощью вычислений в цикле реализована. Задание полностью выполнено.

### **Выводы по заданию.**

Процедура и функция представляют собой подпрограмму. И процедура, и функция - это последовательность операторов, которая имеет имя, список параметров (аргументов) и может быть вызвана из различных частей программы. Отличие заключается в том, что функция в результате своего выполнения возвращает значение, которое может быть использовано там, откуда эта функция вызвана. А процедура не возвращает никакого значения в место вызова [1]. Поэтому вызов функции и вызов процедуры из части программы, выглядит по разному. Вызов функции связан с оператором "=": Параметр = Имя\_функции(Аргумент1, Аргумент2,...). Для вызова процедуры достаточно указать её имя и в скобках перечислить аргументы (параметры). Функции и процедуры могут вызывать сами себя, тогда они называются рекурсивными.



## 2.3 Данные о студенте

Выбрав в меню "Сервис" Задание 2 переходим в обработку "Данные о студенте". Вид окна системы представлен на рисунке 8. Форма при открытии уже содержит данные по умолчанию. При желании, можно ввести новые данные: ФИО и дату рождения студента. Затем нажать на кнопку "Вывод информации о студенте".

The screenshot shows a web browser window with a yellow header bar containing a search bar and navigation icons. Below the header is a yellow bar with a dropdown menu labeled "Сервис". The main content area has a title bar with navigation icons and the text "Задание 2. 'Данные о студенте'". Below the title bar is a button labeled "Еще". The main content area contains a section titled "Текст задания" with the following text: "Проинициализировать три переменные: а) СтудентФИО- записать в переменную строку с полным ФИО студента (например: 'Петров Пётр Петрович'); б) ДатаРожденияСтудента - записать дату рождения; в) ВозрастСтудента - число полных лет. Вывести на экран сообщение в следующем формате : 'Студент Петров П.П. 1980 г.р. 42 года'". Below this text are two input fields: "Введите ФИО полностью:" with the value "Кононова Виктория Юрьевна" and "Введите дату рождения:" with the value "09.10.1984" and a calendar icon. Below the input fields is a button labeled "Вывод информации о студенте". At the bottom of the window is a status bar with a monitor icon and the text "Текущие вызовы: 9 Накопленные вызовы: 30".

Рисунок 8 - Обработка "Данные о студенте"

После нажатия на кнопку "Вывод информации о студенте", внизу окна сообщением будет выведена информация о студенте в заданном формате (рисунок 9).

The screenshot shows a web application window with a yellow header bar containing a search bar and navigation icons. Below the header, there is a section titled "Задание 2. 'Данные о студенте'" with a star icon and a "Еще" button. The main content area is titled "Текст задания" and contains instructions for initializing three variables: StudentFIO (full name), DateOfBirth (date), and Age (age). Below the instructions, there are input fields for "Введите ФИО полностью:" (filled with "Кононова Виктория Юрьевна") and "Введите дату рождения:" (filled with "09.10.1984"). A button labeled "Вывод информации о студенте" is highlighted with a yellow border. At the bottom, a "Сообщения:" section displays a message: "Студент Кононова В.Ю. 1984 г.р. 39 лет(года)". The footer shows "Текущие вызовы: 0" and "Накопленные вызовы: 30".

Рисунок 9 - Вывод информации

Предусмотрена защита от некорректного ввода ФИО: а) если ФИО заполнено пробелами или просто пустое незаполненное поле (рисунок 10); б) если введена только фамилия (рисунок 11); в) если отсутствует отчество (рисунок 12). В этих случаях, пользователь получит соответствующие сообщения на экран.

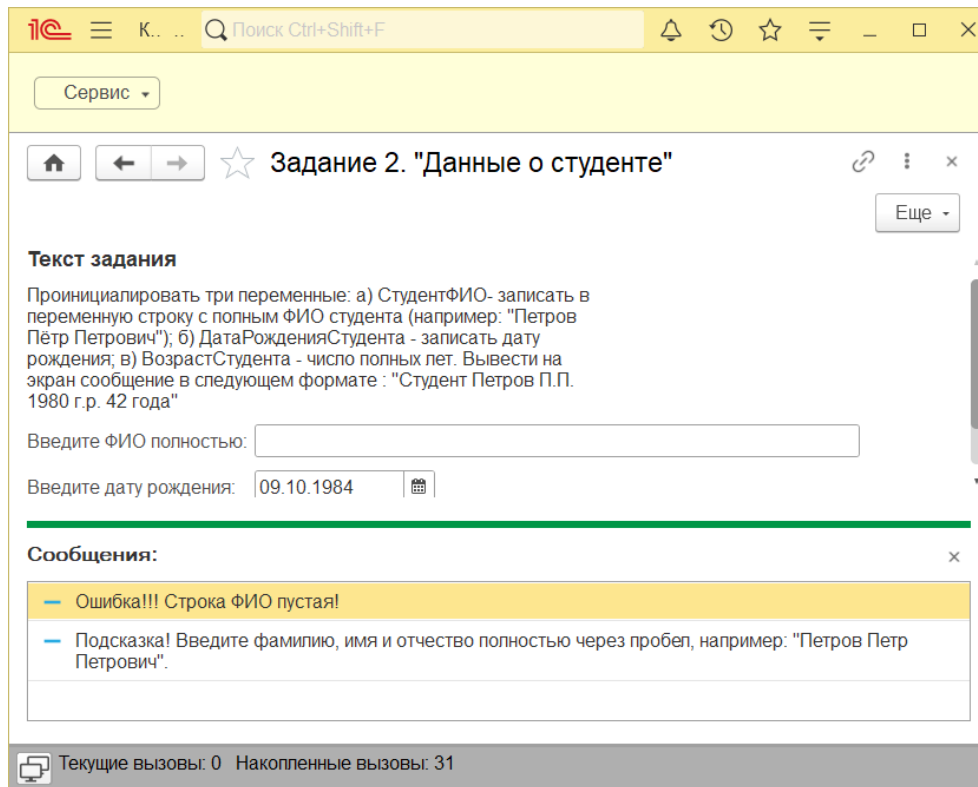


Рисунок 10 - Отсутствуют данные о ФИО

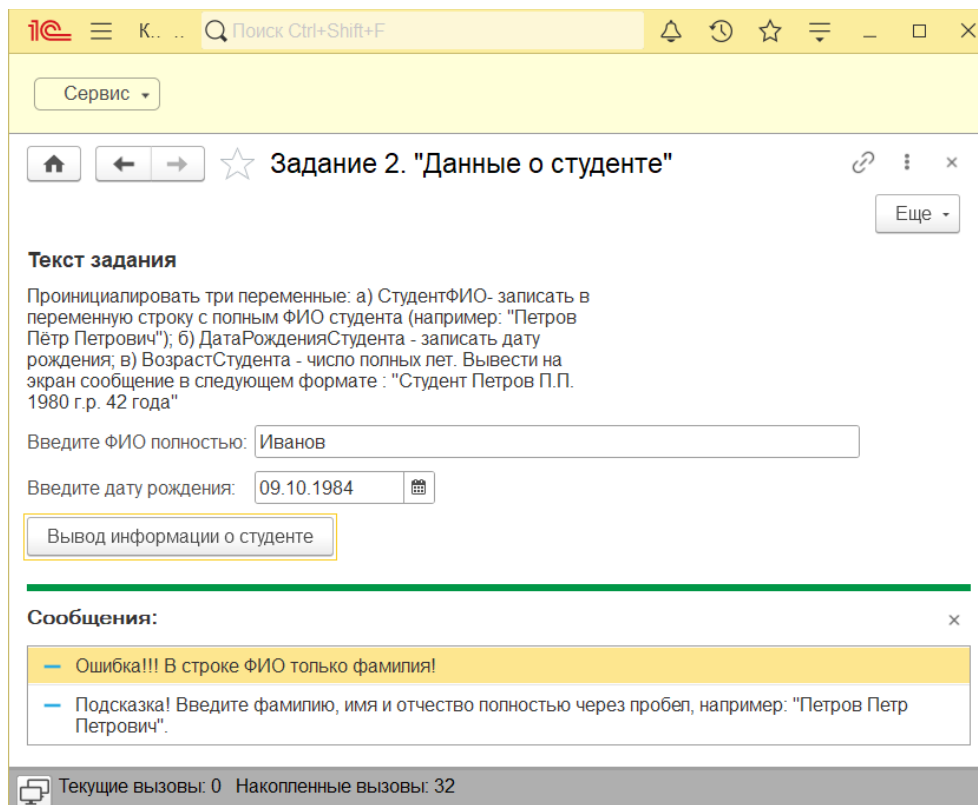


Рисунок 11 - В поле ввода отсутствуют имя и отчество

Сервис

Задание 2. "Данные о студенте"

**Текст задания**

Проинициализировать три переменные: а) СтудентФИО- записать в переменную строку с полным ФИО студента (например: "Петров Петр Петрович"); б) ДатаРожденияСтудента - записать дату рождения; в) ВозрастСтудента - число полных лет. Вывести на экран сообщение в следующем формате : "Студент Петров П.П. 1980 г.р. 42 года"

Введите ФИО полностью:

Введите дату рождения:

**Сообщения:**

- ❌ Ошибка!!! В строке ФИО нет отчества!
- 💡 Подсказка! Введите фамилию, имя и отчество полностью через пробел, например: "Петров Петр Петрович".

Текущие вызовы: 0    Накопленные вызовы: 33

Рисунок 12 - В поле ввода отсутствует отчество

На форму обработки добавлены реквизиты "ДатаРождения", "ФИОПолностью" и команда "ВыводИнформацииОСтуденте" (рисунок 13).

Обработка ДанныеОСтуденте2: Форма

**Форма**

- Командная панель
- Декорация1
- Декорация2
- ФИОПолностью
- ДатаРождения
- ВыводИнформацииОСтуденте

Реквизит	Тип
Объект	(ОбработкаОбъект.Данны...
ДатаРождения	Дата
ФИОПолностью	Строка

Элементы    Командный интерфейс    Реквизиты    Команды    Параметры

**Текст задания**

Проинициализировать три переменные: а) СтудентФИО- записать в переменную строку с полным ФИО студента (например: "Петров Петр Петрович"); б) ДатаРожденияСтудента - записать дату рождения; в) ВозрастСтудента - число полных лет. Вывести на экран сообщение в следующем формате : "Студент Петров П.П. 1980 г.р. 42 года"

Введите ФИО полностью:

Введите дату рождения:

Форма    Модуль

Рисунок 13 - Форма обработки

Код решения, содержащийся в модуле формы, представлен ниже (немного съезжает текст из-за ширины страницы А4 отчета).

&НаКлиенте

Процедура ПриОткрытии(Отказ)

```
СтудентФИО      = "Кононова Виктория Юрьевна";
ДатаРожденияСтудента = Дата("19841009000000");
ФИОПолностью    = СтудентФИО;
ДатаРождения     = ДатаРожденияСтудента;
```

КонецПроцедуры

&НаКлиенте

Процедура ВыводИнформацииОСтуденте(Команда)

```
СтудентФИО      = СокрЛП(ФИОПолностью);
ДатаРожденияСтудента = ДатаРождения;
```

```
Если ФИОВведеноПолностью(СтудентФИО) Тогда
    ВозрастСтудента = ОпределитьВозрастСтудента(ДатаРожденияСтудента);
    ГодРожденияСтудента = Год(ДатаРожденияСтудента);
    СообщитьОСтуденте(СтудентФИО, ГодРожденияСтудента, ВозрастСтудента);
```

Иначе

```
    Сообщить("Подсказка! Введите фамилию, имя и отчество полностью через пробел, например: ""Петров Петр Петрович"".")
```

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Функция ФИОВведеноПолностью(СтудентФИО)

```
РезультатПроверки = Истина;
```

```
Если ПустаяСтрока(СтудентФИО) Тогда
    Сообщить("Ошибка!!! Строка ФИО пустая!");
    РезультатПроверки = Ложь;
```

```
ИначеЕсли ПустаяСтрока(ВыделитьИмяИзПолногоФИО(СтудентФИО)) Тогда
    Сообщить("Ошибка!!! В строке ФИО только фамилия!");
    РезультатПроверки = Ложь;
```

```
ИначеЕсли ПустаяСтрока(ВыделитьОтчествоИзПолногоФИО(СтудентФИО)) Тогда
    Сообщить("Ошибка!!! В строке ФИО нет отчества!");
    РезультатПроверки = Ложь;
```

КонецЕсли;

```
Возврат РезультатПроверки;
```

КонецФункции

&НаКлиенте

Функция ВыделитьФамилиюИзПолногоФИО(СтудентФИО)

НомерПозицииПервогоПробела = СтрНайти(СтудентФИО," ");

Если НомерПозицииПервогоПробела > 0 Тогда

    Фамилия = Лев(СтудентФИО,(НомерПозицииПервогоПробела-1));

Иначе

    Фамилия = СтудентФИО;

КонецЕсли;

Возврат Фамилия;

КонецФункции

&НаКлиенте

Функция ВыделитьИмяИзПолногоФИО(СтудентФИО)

Имя = "";

НомерПозицииПервогоПробела = СтрНайти(СтудентФИО," ");

Если НомерПозицииПервогоПробела > 0 Тогда

    ИОСтудента = СокрЛП(Сред(СтудентФИО,  
(НомерПозицииПервогоПробела+1),СтрДлина(СтудентФИО)));

    НомерПозицииВторогоПробела = СтрНайти(ИОСтудента," ");

    Если НомерПозицииВторогоПробела > 0 Тогда

        Имя = Лев(ИОСтудента, (НомерПозицииВторогоПробела-1) );

    Иначе Имя = ИОСтудента;

    КонецЕсли;

КонецЕсли;

Возврат Имя;

КонецФункции

&НаКлиенте

Функция ВыделитьОтчествоИзПолногоФИО(СтудентФИО)

Отчество = "";

НомерПозицииПервогоПробела = СтрНайти(СтудентФИО," ");

Если НомерПозицииПервогоПробела > 0 Тогда

    ИОСтудента = СокрЛП(Сред(СтудентФИО,  
(НомерПозицииПервогоПробела+1),СтрДлина(СтудентФИО)));

    НомерПозицииВторогоПробела = СтрНайти(ИОСтудента," ");

    Если НомерПозицииВторогоПробела > 0 Тогда

        Отчество = СокрЛП(Сред(ИОСтудента,  
(НомерПозицииВторогоПробела+1), СтрДлина(ИОСтудента)));

    КонецЕсли;

КонецЕсли;

Возврат Отчество;

КонецФункции

&НаКлиенте

Функция ОпределитьВозрастСтудента(ДатаРожденияСтудента)

//В коде ниже определяется количество полных лет студента на текущую дату

ВозрастСтудента = Год(ТекущаяДата()) - Год(ДатаРожденияСтудента);

РазницаВМесяцах = Месяц(ТекущаяДата()) - Месяц(ДатаРожденияСтудента);

Если РазницаВМесяцах < 0 Тогда

ВозрастСтудента = ВозрастСтудента - 1;

ИначеЕсли РазницаВМесяцах = 0 Тогда

ВозрастСтудента = СкорректироватьПоРазницеВДнях(ВозрастСтудента,  
ДатаРожденияСтудента);

КонецЕсли;

Возврат ВозрастСтудента;

КонецФункции

&НаКлиенте

Функция СкорректироватьПоРазницеВДнях(ВозрастСтудента, ДатаРожденияСтудента)

Если (День(ТекущаяДата()) - День(ДатаРожденияСтудента)) < 0 Тогда

ВозрастСтудента = ВозрастСтудента - 1;

КонецЕсли;

Возврат ВозрастСтудента;

КонецФункции

&НаКлиенте

Процедура СообщитьОСтуденте(СтудентФИО, ГодРождения, ВозрастСтудента)

//В коде ниже выводится строка в следующем формате : "Студент Петров П.П. 1980  
г.р. 42 года"

ФамилияСтудента = ВыделитьФамилиюИзПолногоФИО(СтудентФИО);

ИмяСтудента = ВыделитьИмяИзПолногоФИО(СтудентФИО);

ОтчествоСтудента = ВыделитьОтчествоИзПолногоФИО(СтудентФИО);

ИнициалИмениСтудента = Лев(ИмяСтудента,1);

ИнициалОтчестваСтудента = Лев(ОтчествоСтудента,1);

Сообщить("Студент "+ФамилияСтудента+" "

+ИнициалИмениСтудента+"."

+ИнициалОтчестваСтудента+"." "

+Строка(Формат(ГодРождения,"ЧГ=0"))+" г.р. "

+Строка(ВозрастСтудента)+" лет(года)");

КонецПроцедуры

Задание полностью выполнено.

**Выводы по заданию.**

Для работы с примитивными типами данных (Строка, Дата, Число) существует множество функций, их описание можно найти в Синтаксис-помощнике [1]. При выводе текстового сообщения на экран можно скорректировать итоговый вид с помощью конструктора форматной строки, например, вывести год без пробела между цифрами.



## 2.4 Вычисление степени числа

Выбрав в меню "Сервис" Задание 3 переходим в обработку "Вычисление степени числа". Вид окна системы представлен на рисунке 14. Форма при открытии уже содержит данные по умолчанию: число 1 и степень 1. Можно ввести новые данные. Затем нажать на кнопку "Возвести число в степень".

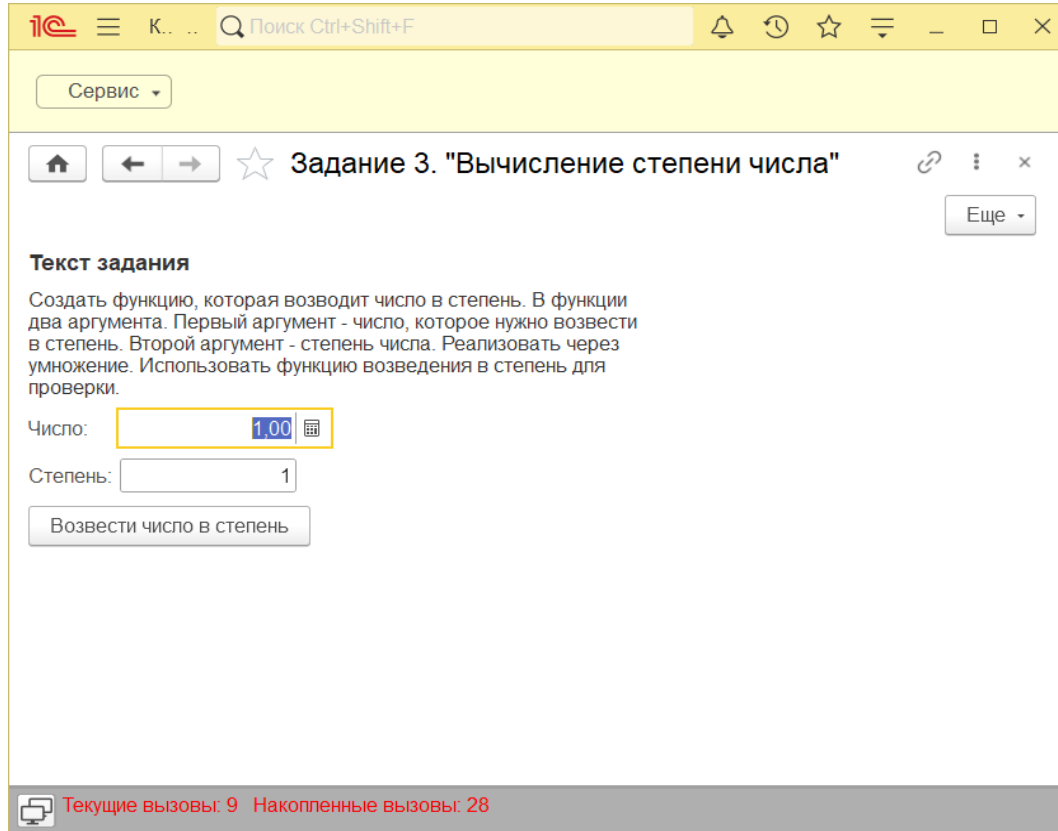


Рисунок 14 - Обработка для вычисления степени числа

Пример возведения числа в степень представлен на рисунке 15. Результат возведения выводится сообщением внизу окна программы.

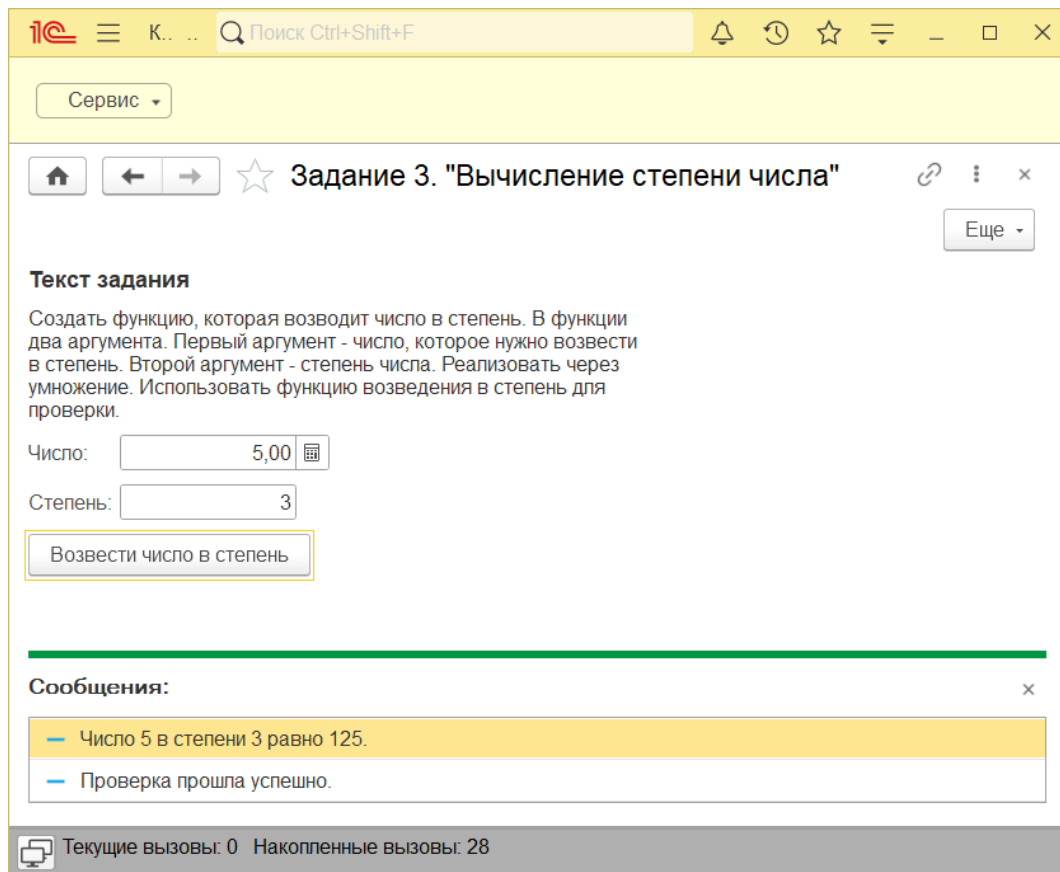


Рисунок 15 - Возведение числа в степень

На форме обработке добавлены два реквизита "НашеЧисло", "Степень" и команда "ВычислитьСтепеньЧисла" (рисунок 16).

Обработка ВычислениеСтепени3: Форма

Форма

- Командная панель
- Декорация1
- Декорация2
- Число
- СтепеньЧисла
- ВычислитьСтепеньЧисла

Реквизит

Реквизит	Тип
Объект	(ОбработкаОбъект.Вычисление_
НашеЧисло	Число
Степень	Число

Элементы Командный интерфейс Реквизиты Команды Параметры

Текст задания

Создать функцию, которая возводит число в степень. В функции два аргумента. Первый аргумент - число, которое нужно возвести в степень. Второй аргумент - степень числа. Реализовать через умножение. Использовать функцию возведения в степень для проверки.

Число:

Степень:

Возвести число в степень

Форма Модуль

Рисунок 16 - Форма обработки вычисления степени

Код решения, содержащийся в модуле формы, представлен ниже (немного съезжает текст из-за ширины страницы А4 отчета).

&НаКлиенте

Процедура ПриОткрытии(Отказ)

НашеЧисло = 1;

Степень = 1;

КонецПроцедуры

&НаКлиенте

Процедура ВозвестиЧислоВСтепень(Команда)

Результат = ВычислитьСтепеньЧисла(НашеЧисло,Степень);

СообщитьРезультат(НашеЧисло,Степень, Результат);

ПроверитьРезультат(НашеЧисло,Степень, Результат);

КонецПроцедуры

&НаКлиенте

Функция ВычислитьСтепеньЧисла(Параметр1,Параметр2)

```
Результат = 1;  
Для Индекс = 1 По Параметр2 Цикл  
    Результат = Результат * Параметр1;  
КонецЦикла;  
Возврат Результат;
```

КонецФункции

&НаКлиенте

Процедура СообщитьРезультат(НашеЧисло,Степень, Результат)

```
Сообщить("Число " + НашеЧисло + " в степени " + Степень + " равно " + Результат +  
".");
```

КонецПроцедуры

&НаКлиенте

Процедура ПроверитьРезультат(НашеЧисло,Степень, Результат)

```
Если Результат = Pow(НашеЧисло,Степень) Тогда  
    Сообщить("Проверка прошла успешно.");  
Иначе  
    Сообщить("Результат не прошел проверку!");  
КонецЕсли;
```

КонецПроцедуры

Задание выполнено полностью.

### Выводы по заданию.

Оператор цикла "Для" предназначен для циклического повторения операторов, находящихся внутри конструкции "Цикл – КонецЦикла".

Синтаксис:

```
Для <Имя_переменной> = <Выражение 1> По <Выражение 2> Цикл  
    // Операторы  
КонецЦикла;
```

Перед началом выполнения цикла значение "Выражение 1" присваивается переменной "Имя\_переменной". Значение "Имя\_переменной" автоматически увеличивается при каждом проходе цикла. Величина приращения счетчика при каждом выполнении цикла равна 1.

Цикл выполняется, пока значение переменной "Имя\_переменной" меньше или равно значению "Выражение 2". Цикл "Для" - цикл с предусловием: условие выполнения цикла всегда проверяется в начале, перед выполнением цикла.

## 2.5 Демонстрация работы функции определения типа данных

Выбрав в меню "Сервис" Задание 4 переходим в обработку "Демонстрация работы функции определения типа данных" (рисунок 17). Предусмотрено семь параметров разных типов для ввода: Строка, Число, Дата, Истина, Ложь, NULL, произвольный. По умолчанию все поля уже заполнены необходимой информацией. Можно просто нажать на кнопку "Определить тип данных". После нажатия кнопки будут выведены сообщения о типах данных внизу окна программы (рисунок 18).

Сервис ▾

Задание 4. "Демонстрация работы функции определения типа данных" Еще ▾

**Текст задания**

Написать функцию, которая в зависимости от типа переданного параметра выводит сообщение.  
Если типы "число", "строка", "дата", то сообщение: "Данный параметр имеет тип " и далее название типа.  
Если тип "булево", то сообщение: "Данный параметр может принимать значение "истина" или "ложь"".   
Если другой тип, то сообщение: "тип не определен".

Введите строку (Параметр1):

Введите число (Параметр2):

Введите дату (Параметр3):

Поставьте галочку (Параметр4): ☒

Не ставьте галочку (Параметр5): ☐

Здесь NULL (Параметр6):

Здесь неопределено (Параметр7):

Текущие вызовы: 0    Накопленные вызовы: 31

Рисунок 17 - Обработка "Демонстрация работы функции определения типа данных"

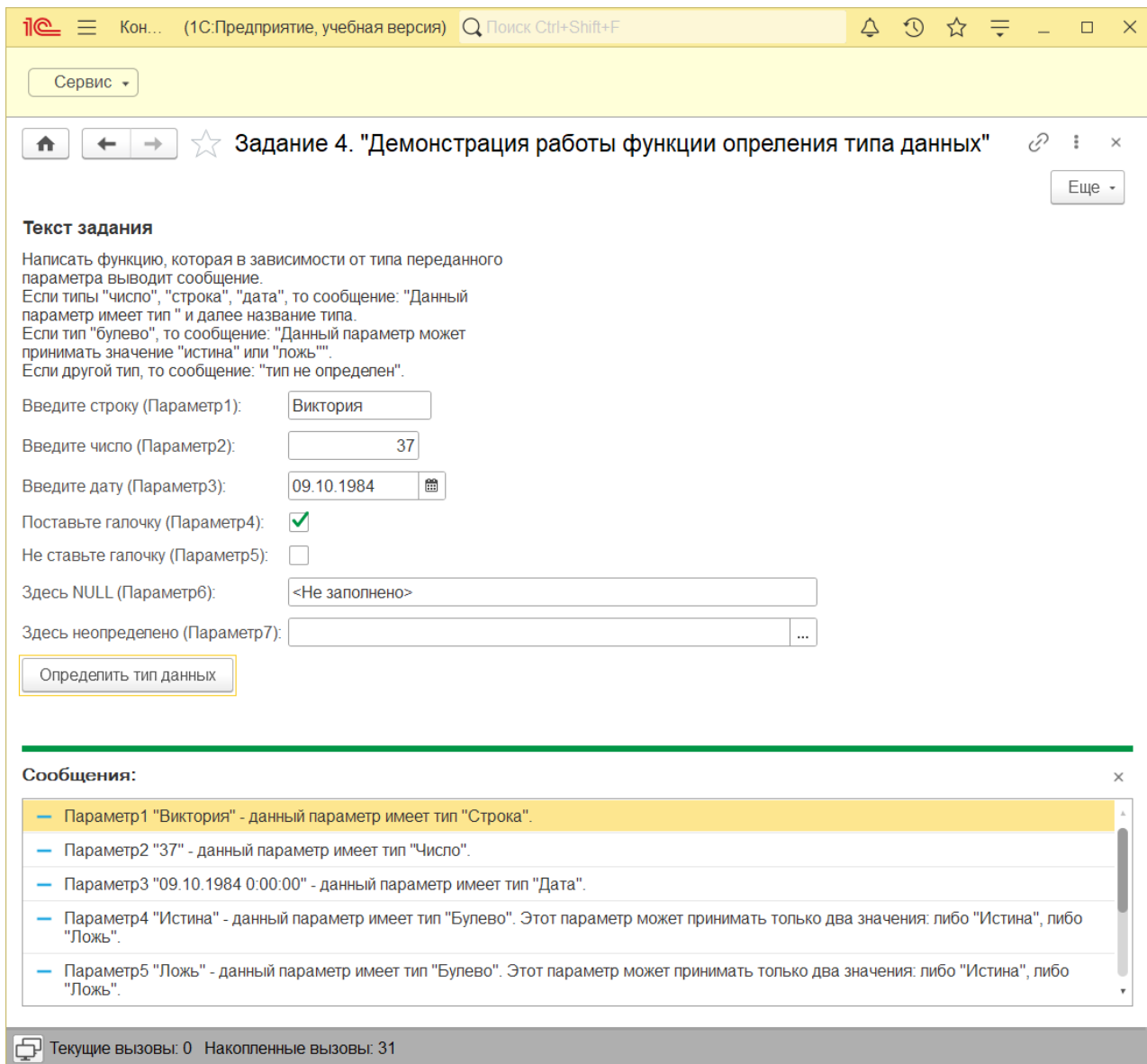


Рисунок 18 - Вывод сообщений о типах данных всех параметров

На форму обработки (рисунок 19) добавлены кнопка "ОпределитьТипДанных" и семь реквизитов разных типов (таблица 1).

Рисунок 19 - Форма обработки

Таблица 1 - Реквизиты, добавленные на форму обработки

Реквизит	Тип	Назначение
ВведитеСтроку	Строка	Параметр1
ВведитеЧисло	Число	Параметр2
ВведитеДату	Дата	Параметр3
НеСтавьтеГалочку	Булево	Параметр4
ПоставьтеГалочку	Булево	Параметр5
ЗдесьNULL	Null	Параметр6
ЗдесьНеопределено	Произвольный	Параметр7

Код решения, содержащийся в модуле формы, представлен ниже (немного съезжает текст из-за ширины страницы А4 отчета).

&НаКлиенте

Процедура ПриОткрытии(Отказ)

```

ВведитеСтроку = "Виктория";
ВведитеЧисло = 37;
ВведитеДату = Дата("19841009000000");
ПоставьтеГалочку = Истина;

```

```
НеСтавьтеГалочку = Ложь;  
ЗдесьNULL = NULL;  
ЗдесьНеопределено = Неопределено;
```

КонецПроцедуры

&НаКлиенте

Процедура ОпределитьТипДанных(Команда)

```
Параметр1 = ВведитеСтроку;  
Параметр2 = ВведитеЧисло;  
Параметр3 = ВведитеДату;  
Параметр4 = ПоставьтеГалочку;  
Параметр5 = НеСтавьтеГалочку;  
Параметр6 = ЗдесьNULL;  
Параметр7 = ЗдесьНеопределено;
```

```
СообщитьТипПараметра(Параметр1, 1);  
СообщитьТипПараметра(Параметр2, 2);  
СообщитьТипПараметра(Параметр3, 3);  
СообщитьТипПараметра(Параметр4, 4);  
СообщитьТипПараметра(Параметр5, 5);  
СообщитьТипПараметра(Параметр6, 6);  
СообщитьТипПараметра(Параметр7, 7);
```

КонецПроцедуры

&НаКлиенте

Процедура СообщитьТипПараметра(Параметр, Номер)

```
Если Строка(ТипЗнч(Параметр)) = "Булево" Тогда  
    Сообщить("Параметр" + Номер + " "" + Формат(Параметр, "БЛ=Ложь;  
БИ=Истина") + """" + " - данный параметр имеет тип """" + Строка(ТипЗнч(Параметр)) + """".  
    Этот параметр может принимать только два значения: либо ""Истина"", либо ""Ложь"".");  
Иначе  
    Сообщить("Параметр" + Номер + " "" + Формат(Параметр, "БЛ=Ложь;  
БИ=Истина") + """" + " - данный параметр имеет тип """" + Строка(ТипЗнч(Параметр)) +  
    """".");  
КонецЕсли;
```

КонецПроцедуры

Задание полностью выполнено.

### **Выводы по заданию.**

На форме можно создавать реквизиты различных типов, поля для ввода значений уже подстроены под определенный тип, например, поле для даты, содержит мини-календарь. Предусмотрена защита от некорректного ввода в поля определенного типа.

С помощью функции ТипЗнч(Параметр) можно получить тип значения произвольного параметра. А с помощью "Конструктора форматной строки..." настроить формат вывода параметра в сообщении.



## 2.6 Демонстрация исключений

Выбрав в меню "Сервис" Задание 5 переходим в обработку "Демонстрация исключений" (рисунок 20). Поля на форме по умолчанию заполнены данными, вызывающими исключительные ситуации. Рассмотрены три самых распространенных случая возникновения исключительных ситуаций: 1) исключение, возникающее при делении на ноль; 2) исключение, возникающее при извлечении квадратного корня из отрицательного числа; 3) исключение, возникающее при попытке преобразовании буквенного текста с типом "Строка" в тип "Число".

1e ≡ Конфигурация (1С:Предприятие, учебная версия)

Сервис ▾

🏠 ⬅ ➡ ☆ Задание 5. "Демонстрация исключений"

**Текст задания**

Самостоятельно ознакомиться с конструкцией Попытка – Исключение – КонецПопытки. Привести примеры использования.

**Исключение "Деление на ноль"**

Делимое:

Делитель:

Вычислить частное (ловить исключение)

Вычислить частное (без ловли исключений)

**Исключение "Корень из отрицательного числа"**

Число:

Вычислить корень (ловить исключение)

Вычислить корень (без ловли исключений)

**Исключение при преобразовании к типу "Число"**

Строка:

Преобразовать к типу "Число" (ловить исключение)

Преобразовать к типу "Число" (без ловли исключений)

Рисунок 20 - Обработка демонстрация исключений

Для деления 5 на ноль нужно нажать кнопку "Вычислить частное (ловить исключение)". Сработает код, содержащий конструкцию Попытка-Исключение-КонецПопытки.

&НаКлиенте

Процедура ВычислитьЧастноеЛовитьИсключение(Команда)

Попытка

Результат = Делимое/Делитель;

Сообщить("Частное = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры

Исключение будет поймано и внизу окна появятся сообщения с информацией об ошибке (рисунок 21).

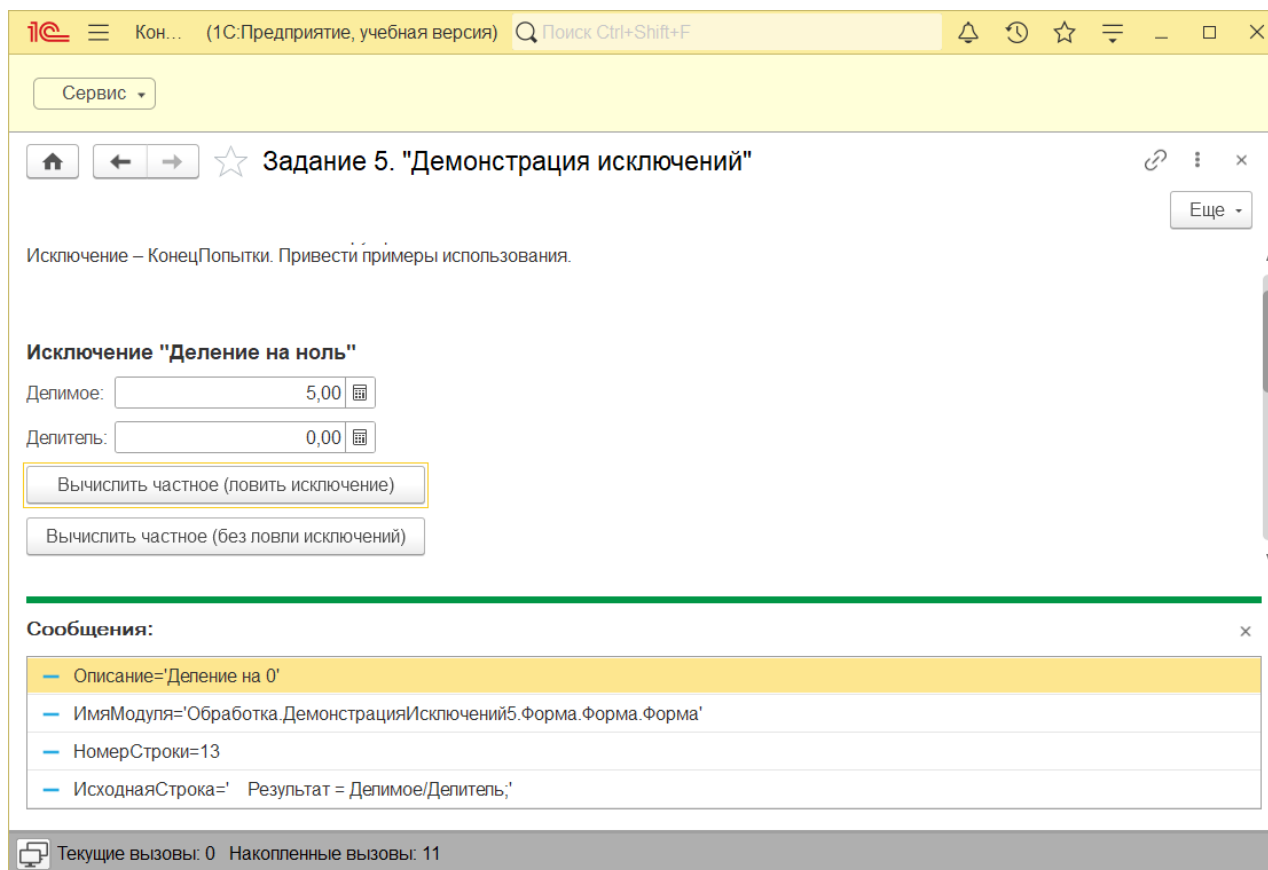


Рисунок 21 - Сообщения об ошибке при делении на ноль

При нажатии на кнопку "Вычислить частное (без ловили исключений)" сработает код, представленный ниже (не содержит конструкцию Попытка-Исключение-КонецПопытки).

```
&НаКлиенте  
Процедура ВычислитьЧастное(Команда)  
    Результат = Делимое/Делитель;  
    Сообщить("Частное = " + Результат + ".");  
КонецПроцедуры
```

В результате на экране появится отдельное окно с текстом "Деление на 0" (рисунок 22)

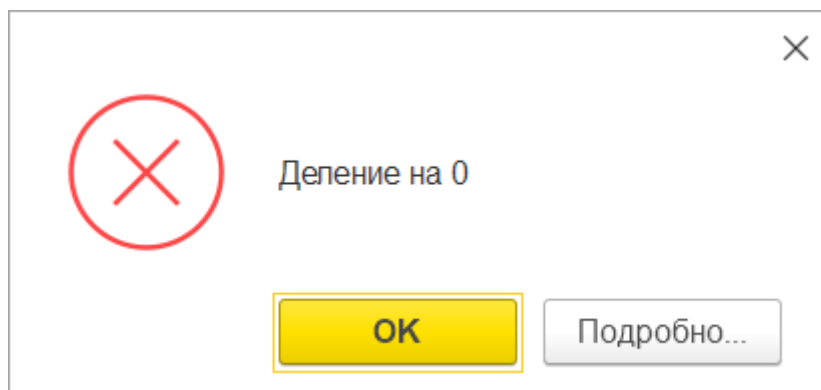


Рисунок 22 - Окно с текстом "Деление на 0"

После нажатия на кнопку "Подробнее..." можно просмотреть информацию об ошибке, используя полосу прокрутки (рисунок 23).

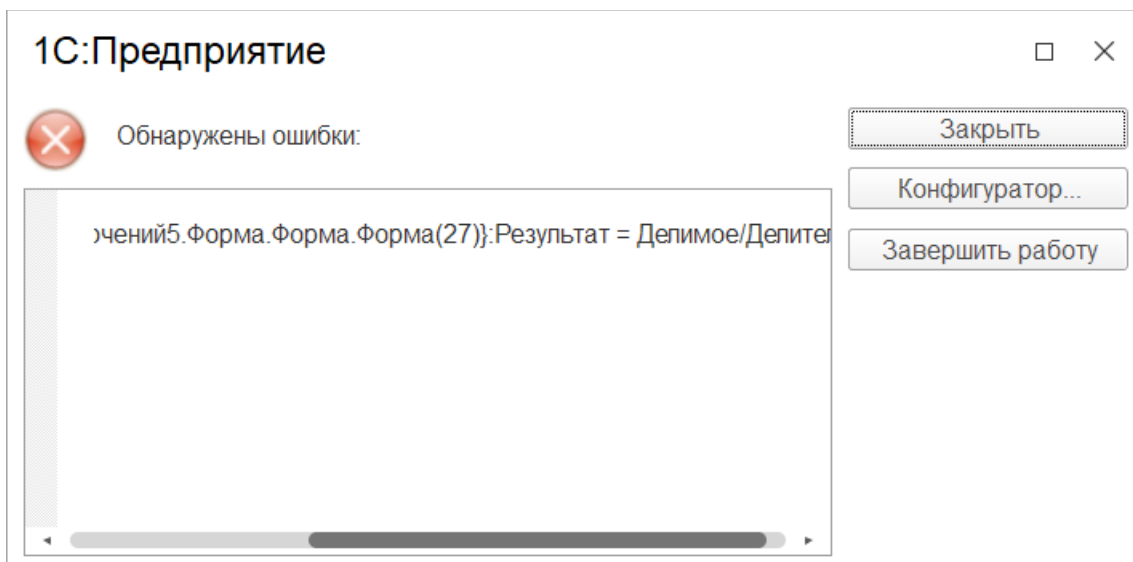


Рисунок 23 - Просмотр информации об ошибке

Для вычисления квадратного корня из отрицательного числа -25 нужно нажать кнопку "Вычислить корень (ловить исключение)". Сработает код, содержащий конструкцию Попытка-Исключение-КонецПопытки.

&НаКлиенте

Процедура ВычислитьКореньЛовитьИсключение(Команда)

Попытка

Результат = Sqrt(Число);

Сообщить("Корень = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры

Исключение будет поймано и внизу окна появятся сообщения с информацией об ошибке извлечения корня из отрицательного числа (рисунок 24).

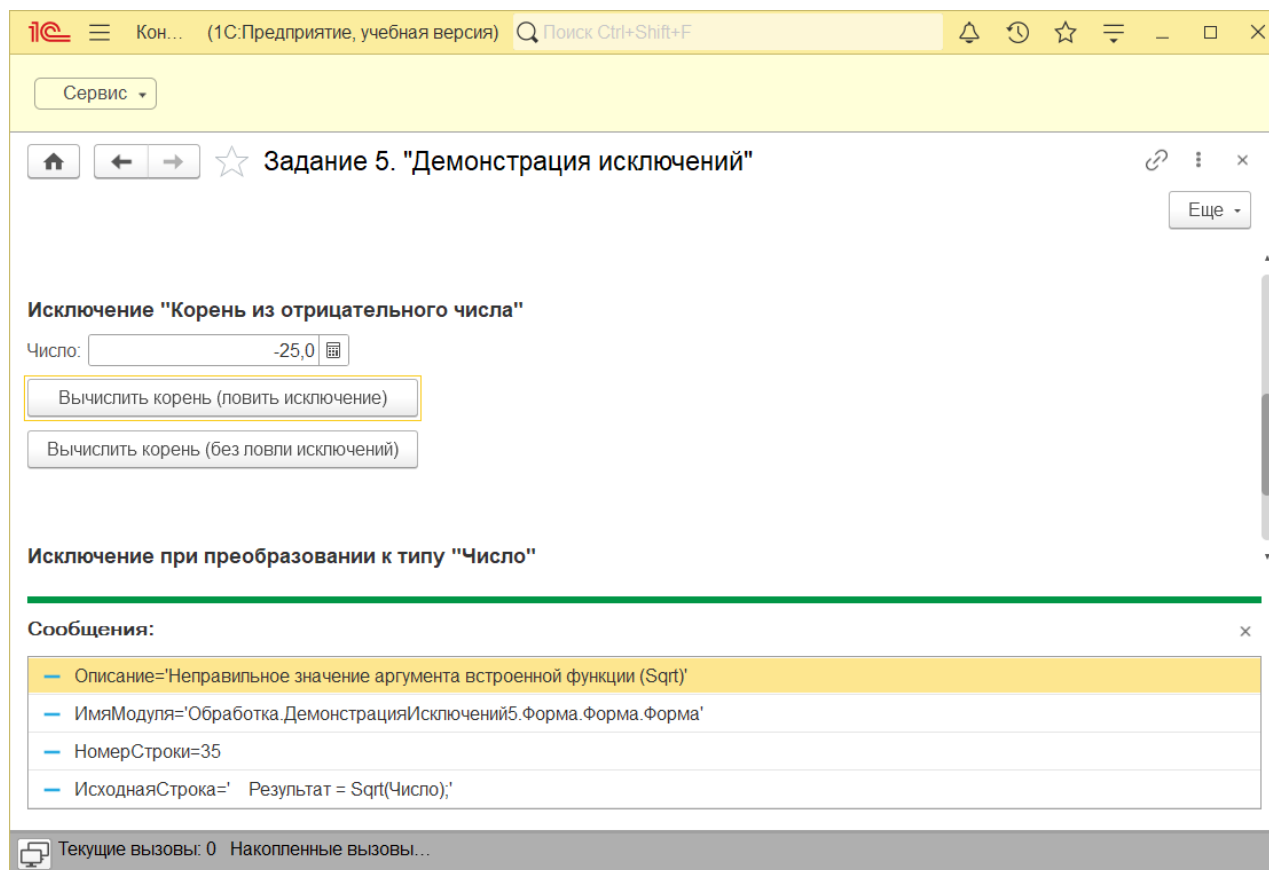


Рисунок 24 - Сообщения с информацией об ошибке извлечения корня из отрицательного числа

При нажатии на кнопку "Вычислить корень (без ловили исключений)" сработает код, представленный ниже (не содержит конструкцию Попытка-Исключение-КонецПопытки).

&НаКлиенте

Процедура ВычислитьКорень(Команда)

    Результат = Sqrt(Число);

    Сообщить("Корень = " + Результат + ".");

КонецПроцедуры

В результате на экране появится отдельное окно (рисунок 25).

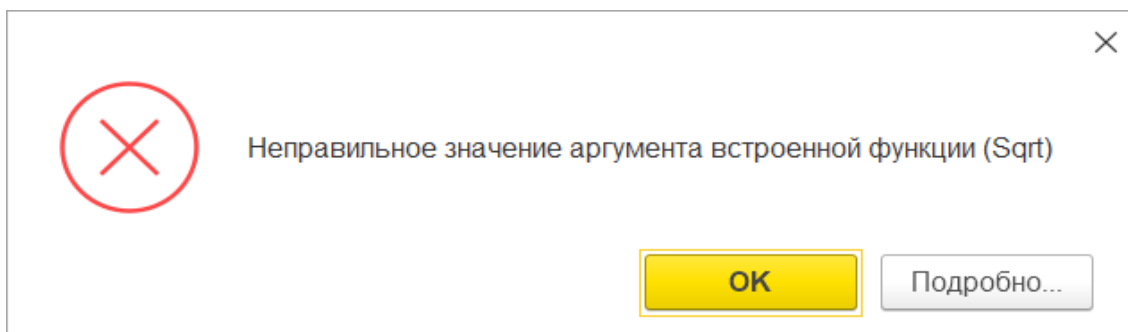


Рисунок 25 - Окно с сообщением об ошибке

После нажатия на кнопку "Подробнее..." можно просмотреть информацию об ошибке, используя полосу прокрутки (рисунок 26).

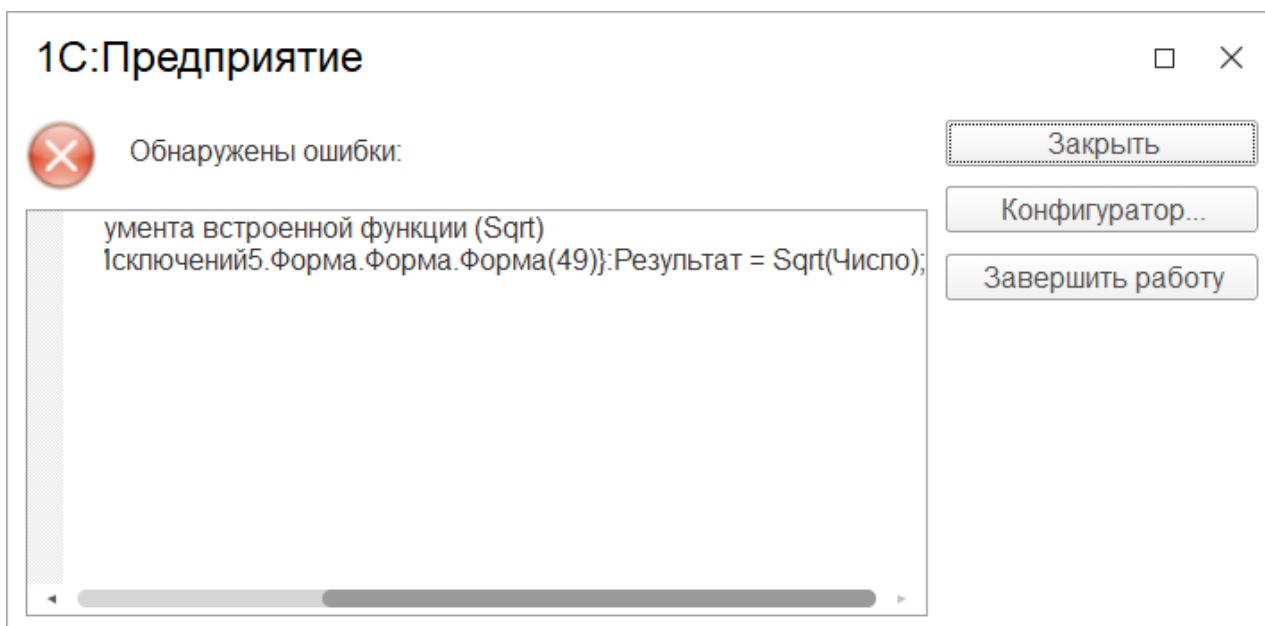


Рисунок 26 - Информация об ошибке

Для преобразования текста "Привет мир" к типу "Число" нужно нажать кнопку "Преобразовать к типу "Число" (ловить исключение)". Сработает код, содержащий конструкцию Попытка-Исключение-КонецПопытки.

&НаКлиенте

Процедура ПреобразоватьКТипуЧислоЛовитьИсключение(Команда)

Попытка

Результат = Число(Строка);

Сообщить("Строка преобразована в Число = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры

Исключение будет поймано и внизу окна появятся сообщения с информацией об ошибке во время преобразования (рисунок 27).

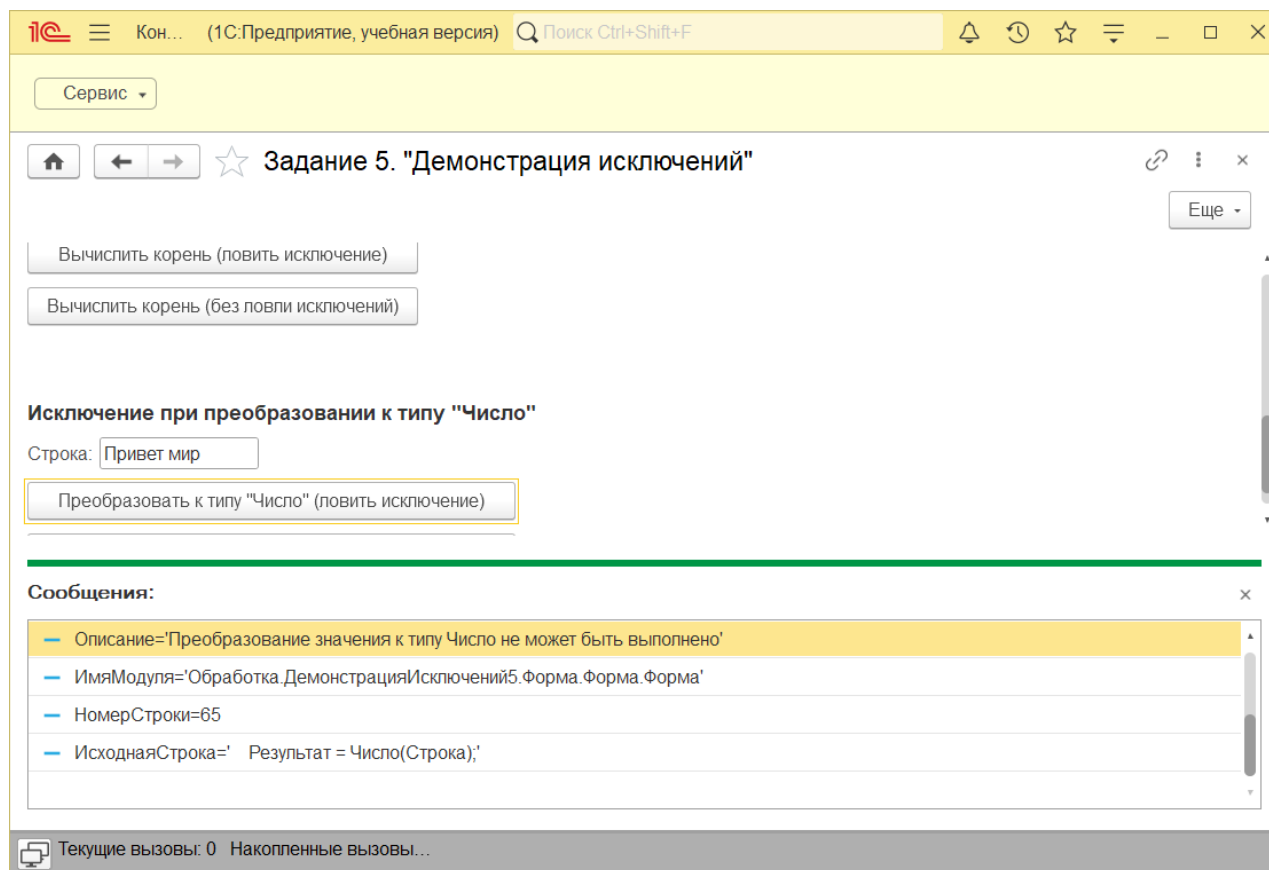


Рисунок 27 - Вывод сообщений с информацией об ошибке

При нажатии на кнопку "Преобразовать к типу "Число" (без ловли исключения)" сработает код, представленный ниже (не содержит конструкцию Попытка-Исключение-КонецПопытки).

&НаКлиенте

Процедура ПреобразоватьКТипуЧисло(Команда)

Результат = Число(Строка);

Сообщить("Строка преобразована в Число = " + Результат + ".");

КонецПроцедуры

В результате на экране появится отдельное окно (рисунок 28).

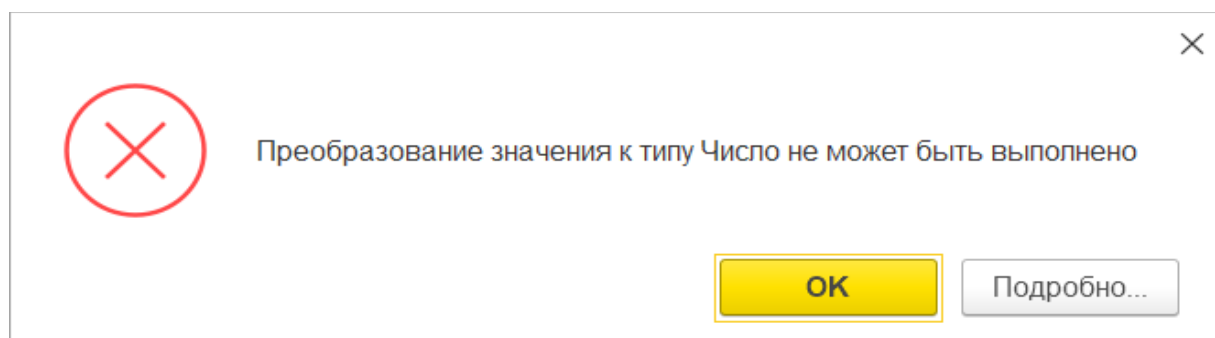


Рисунок 28 - Окно с сообщением об ошибке

После нажатия на кнопку "Подробнее..." можно просмотреть информацию об ошибке, используя полосу прокрутки (рисунок 29).

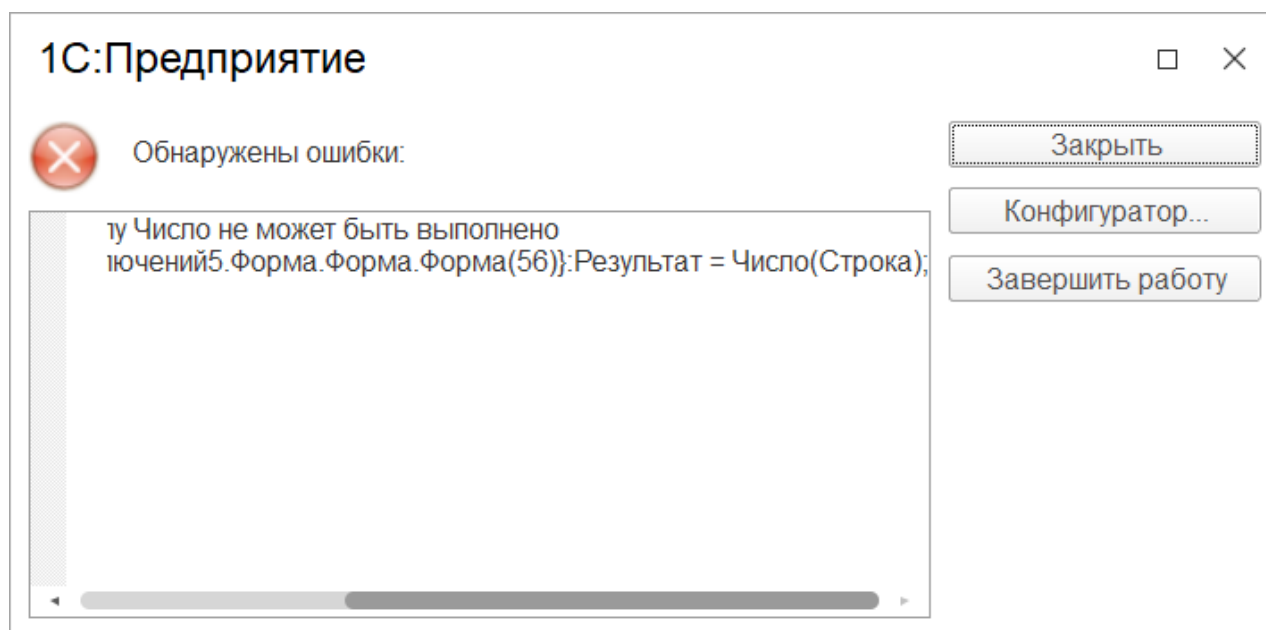


Рисунок 29 - Информация об ошибке

На форме обработки демонстрации исключений (рисунок 30) были созданы реквизиты Делимое, Делитель, Строка и Число и шесть команд, соответствующих кнопкам на форме.

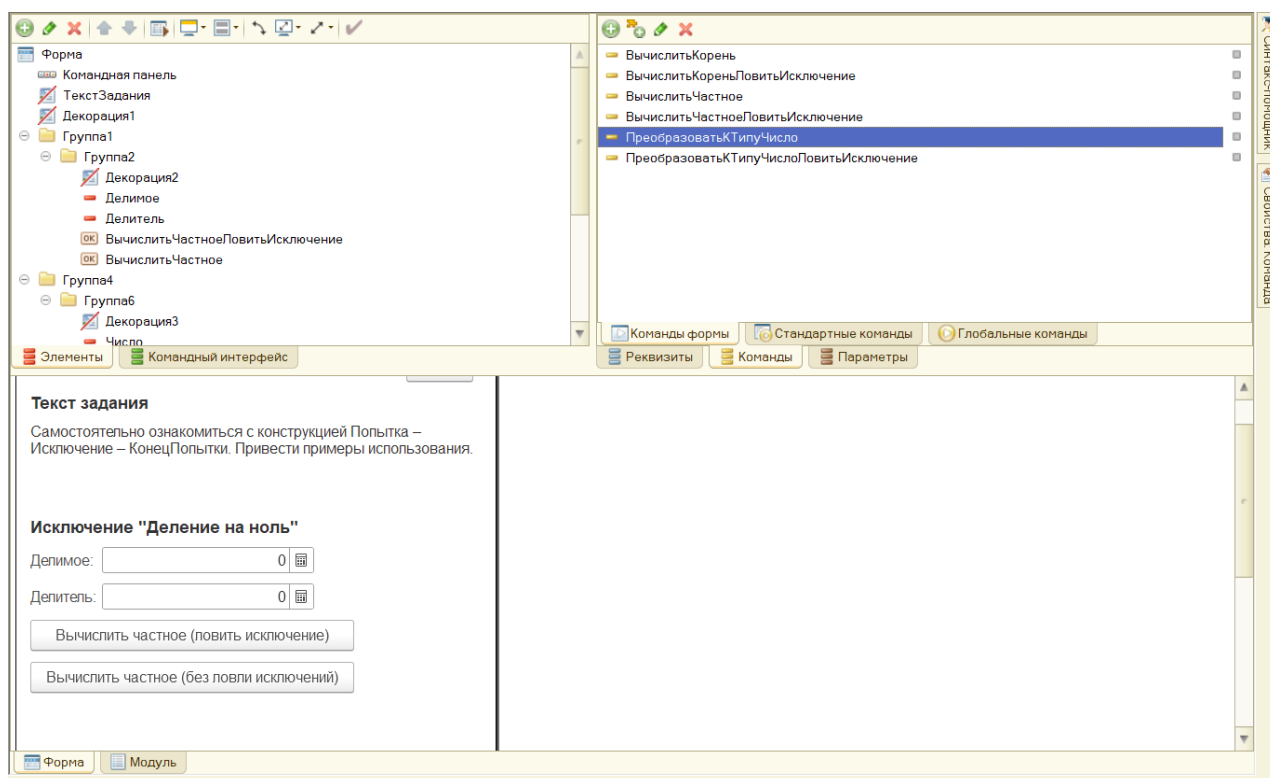


Рисунок 30 - Форма обработки демонстрации исключений

Код решения, содержащийся в модуле формы, представлен ниже.

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Делимое = 5;

Делитель = 0;

Число = -25;

Строка = "Привет мир!"

КонецПроцедуры

&НаКлиенте

Процедура ВычислитьЧастноеЛовитьИсключение(Команда)

Попытка

Результат = Делимое/Делитель;

Сообщить("Частное = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры



&НаКлиенте

Процедура ВычислитьЧастное(Команда)

Результат = Делимое/Делитель;

Сообщить("Частное = " + Результат + ".");

КонецПроцедуры

&НаКлиенте

Процедура ВычислитьКореньЛовитьИсключение(Команда)

Попытка

Результат = Sqrt(Число);

Сообщить("Корень = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры

&НаКлиенте

Процедура ВычислитьКорень(Команда)

Результат = Sqrt(Число);

Сообщить("Корень = " + Результат + ".");

КонецПроцедуры

&НаКлиенте

Процедура ПреобразоватьКТипуЧисло(Команда)

Результат = Число(Строка);

Сообщить("Строка преобразована в Число = " + Результат + ".");

КонецПроцедуры

&НаКлиенте

Процедура ПреобразоватьКТипуЧислоЛовитьИсключение(Команда)

Попытка

Результат = Число(Строка);

Сообщить("Строка преобразована в Число = " + Результат + ".");

Исключение

Инфо = ИнформацияОбОшибке();

Сообщить("Описание=" + Инфо.Описание + "");

Сообщить("ИмяМодуля=" + Инфо.ИмяМодуля + "");

Сообщить("НомерСтроки=" + Инфо.НомерСтроки);

Сообщить("ИсходнаяСтрока=" + Инфо.ИсходнаяСтрока + "");

КонецПопытки;

КонецПроцедуры

## Выводы по заданию.

В общем случае не рекомендуется перехватывать исключения [2]. В частности не нужно перехватывать исключения только ради выдачи сообщения об ошибке. Необработанное исключение в любом случае будет выдано пользователю в виде сообщения об ошибке (а также будет автоматически записано в журнал регистрации для администратора и отправлено в сервис регистрации ошибок).

Тем не менее, необходимость перехвата исключений в коде все же возникает. Например, для того чтобы уточнить текст ошибки, дополнив его прикладной, понятной конечному пользователю, информацией. Однако при этом необходимо фиксировать причину ошибки в журнале регистрации для того, чтобы системный администратор имел возможность выполнить диагностику проблемы и при необходимости передать информацию об ошибке в службу технической поддержки.

При этом рекомендуется записывать в журнал регистрации подробное представление исключения, а краткое представление добавлять в текст сообщения пользователю.

Неправильно использовать исключения для приведения значения к типу. Для таких операций необходимо использовать возможности объекта ОписаниеТипов.

Например, неправильно:

```
Попытка
КоличествоДнейРазрешения = Число (Значение) ;
Исключение
КоличествоДнейРазрешения = 0; // значение по умолчанию
КонецПопытки;
```

Правильно:

```
ОписаниеТипа = Новый ОписаниеТипов ("Число") ;
КоличествоДнейРазрешения = ОписаниеТипа.ПривестиЗначение (Значение) ;
```

Реализованная в системе 1С:Предприятие стратегия работы с исключениями ориентирована на то, что разработчик в большинстве случаев не обрабатывает исключения во встроенном языке. Независимо от характера исключения система автоматически завершит текущее действие и отобразит ошибку в стандартном окне. Если исключение восстановимое, то работа будет продолжена и пользователь сможет повторить вызванное действие, например, снова нажав кнопку в форме. Если исключение невосстановимое, то работа системы будет завершена.

Рекомендуется без необходимости не обрабатывать исключения в модулях и не дублировать те проверки, которые делает система и сама генерирует по ним исключения, а предоставить системе возможность отработать их стандартным способом.

## 2.7 Демонстрация изменений значений параметров в функции

В синтаксис-помощнике есть описание, что такое "Функция" (рисунок 31).

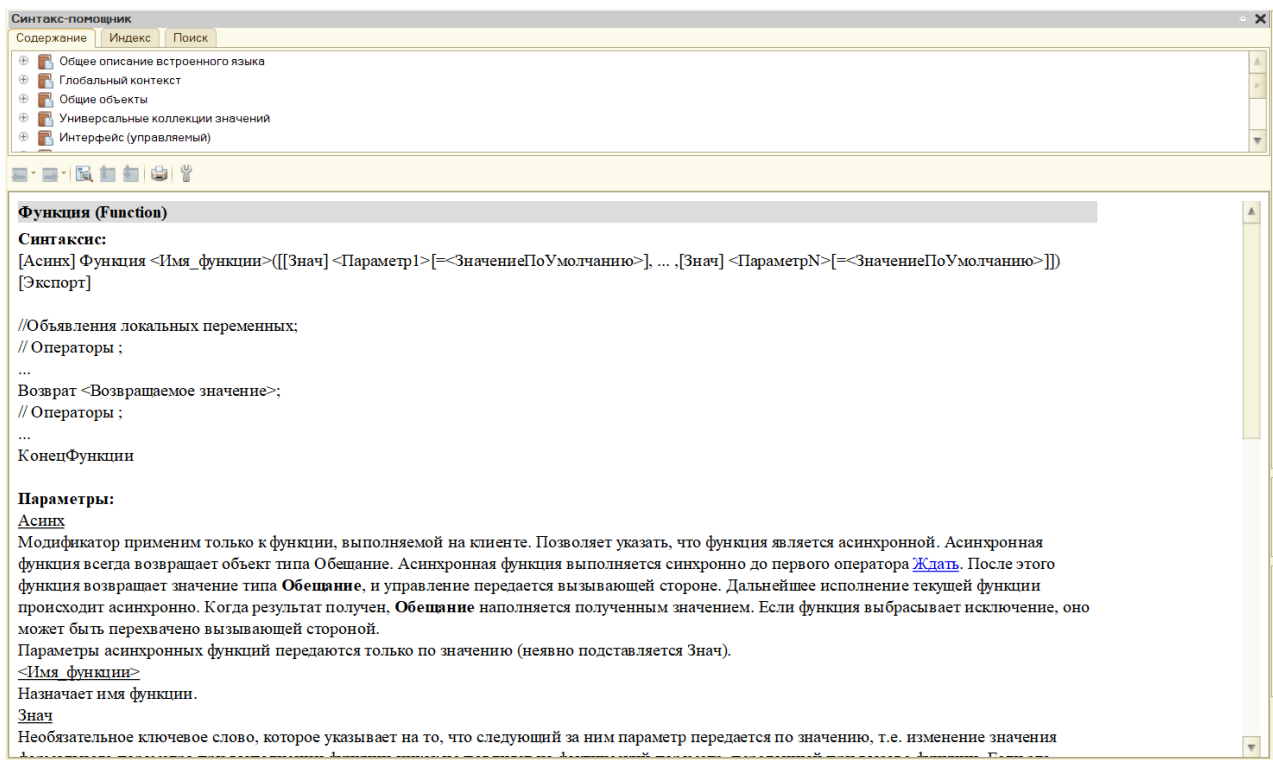


Рисунок 31 - Описание функции в синтаксис-помощнике

Из описания, можно узнать, что перед именем параметра можно указывать ключевое слово "Знач". Если ключевое слово "Знач" не указано, то параметр функции передается по ссылке, то есть изменение внутри функции значения формального параметра приведет к изменению значения соответствующего фактического параметра [1].

Функцию можно сделать "общей" и использовать в разных программных модулях, добавив в описание ключевое слово "Экспорт". Если в модуле обычного, управляемого приложения, внешнего соединения, сеанса или общем программном модуле в теле описания функции использовано ключевое слово "Экспорт", то это означает, что данная функция является доступной из всех других программных модулей конфигурации [1].

Для демонстрации использования ключевого слова "Экспорт" на дереве конфигурации был добавлен общий модуль "УчебныйМодуль" (рисунок 32).

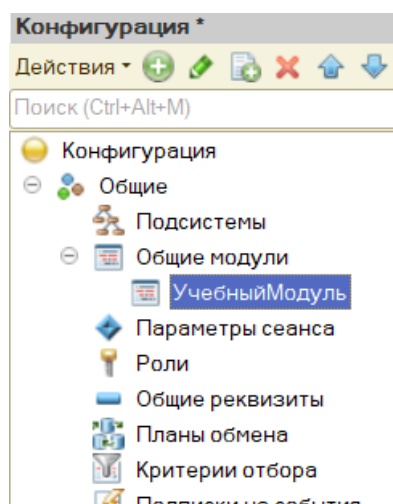


Рисунок 32 - Добавление общего модуля

В свойствах Учебного модуля выбрана опция "Клиент" (рисунок 33). Если это свойство установлено в значение Истина, то процедуры и функции общего модуля могут быть использованы в толстом клиенте в режиме обычного приложения [1].

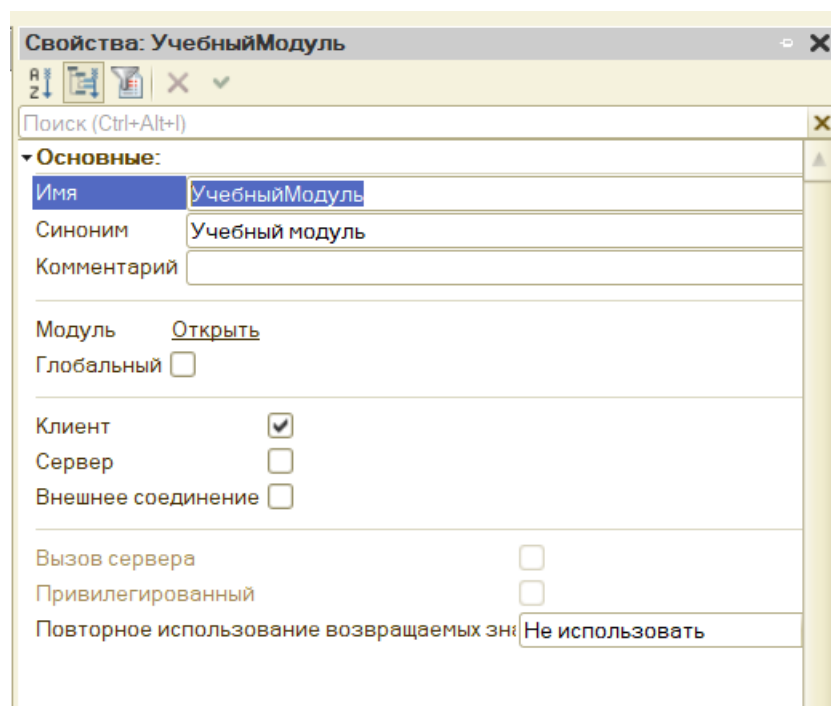


Рисунок 33 - Свойства Учебного модуля

В учебном модуль были добавлены две функции для вычисления квадрата аргумента. В описании первой функции используется ссылка (указатель) на Аргумент и в процессе вычислений исходное значение Аргумента в вызывающем коде будет изменено. В описании второй функции использовано ключевое слово "Знач". Текст учебного модуля представлен ниже.

&НаКлиенте

Функция ВозведениеВКвадрат(Аргумент) Экспорт

Аргумент = Аргумент \* Аргумент;

Возврат Аргумент;

КонецФункции

&НаКлиенте

Функция ВозведениеВКвадратЗначения(Знач Аргумент) Экспорт

Аргумент = Аргумент \* Аргумент;

Возврат Аргумент;

КонецФункции

Для демонстрации изменения значения в процессе работы функции была создана обработка ДемонстрацияИзмененийЗначенийПараметровВФункции6 (рисунок 34).

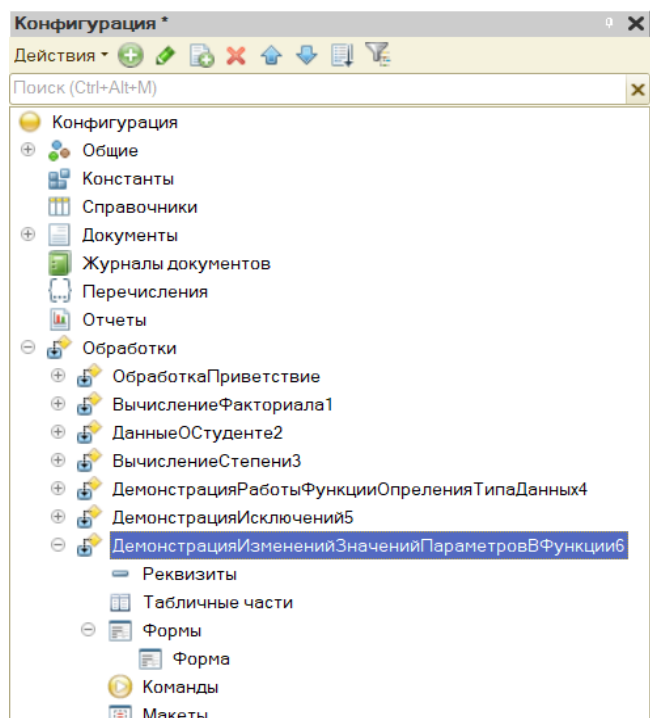


Рисунок 34 - Добавление обработки  
ДемонстрацияИзмененийЗначенийПараметровВФункции6

На форму обработки (рисунок 35) добавлены реквизиты Параметр, Параметр2, Результат, Результат2 и две команды ВозвестиВКвадрат, ВозвестиВКвадратЗначение.

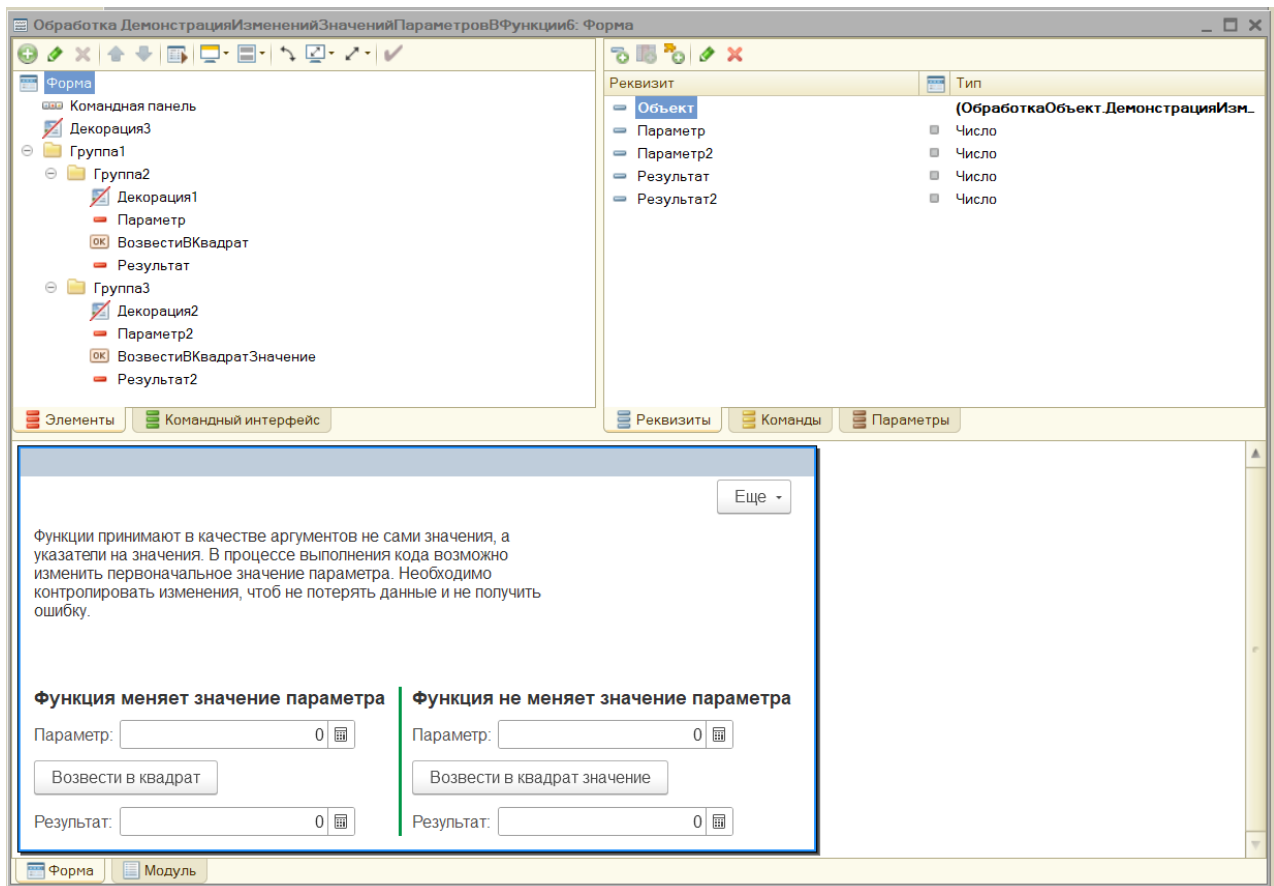


Рисунок 35 - Форма обработки

Код модуля формы представлен ниже

&НаКлиенте

Процедура ВозвестиВКвадрат(Команда)

```
Сообщить("Параметр = " + Параметр);
Результат = УчебныйМодуль.ВозведениеВКвадрат(Параметр);
Сообщить("Параметр " + Параметр + " в квадрате равен " + Результат);
```

КонецПроцедуры

&НаКлиенте

Процедура ВозвестиВКвадратЗначение(Команда)

```
Сообщить("Параметр = " + Параметр2);
Результат2 = УчебныйМодуль.ВозведениеВКвадратЗначения(Параметр2);
Сообщить("Параметр " + Параметр2 + " в квадрате равен " + Результат2);
```

КонецПроцедуры

В режиме пользователя вводим в поле Параметр значение, например, 5 (рисунок 36).

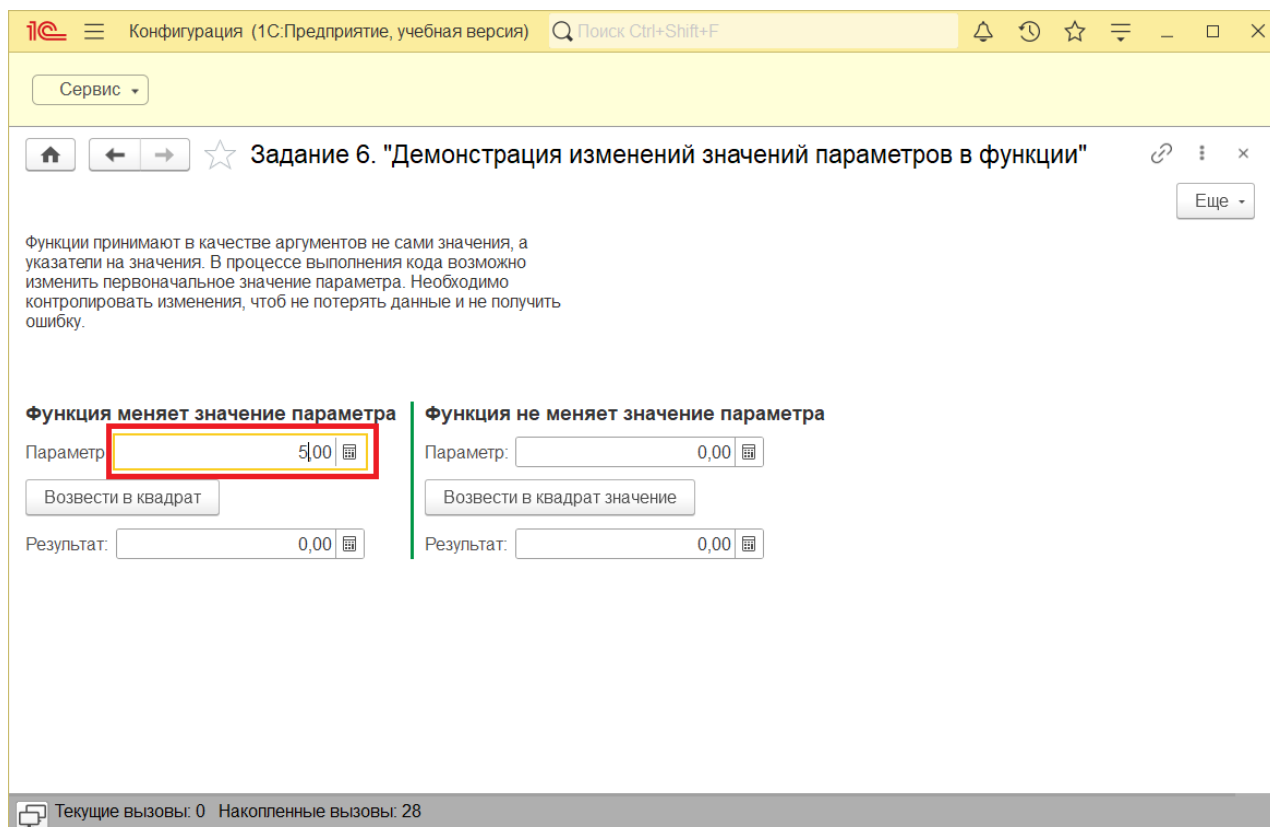


Рисунок 36 - Ввод значений параметра

Затем нажимаем кнопку "Возвести в квадрат". Будет вызвана функция из учебного модуля. Текст функции представлен ниже.

&НаКлиенте

Функция ВозведениеВКвадрат(Аргумент) Экспорт

Аргумент = Аргумент \* Аргумент;

Возврат Аргумент;

КонецФункции

В описании функции отсутствует ключевое слово "Знач", поэтому исходное значение параметра будет изменено: заменено на квадрат (рисунок 37).

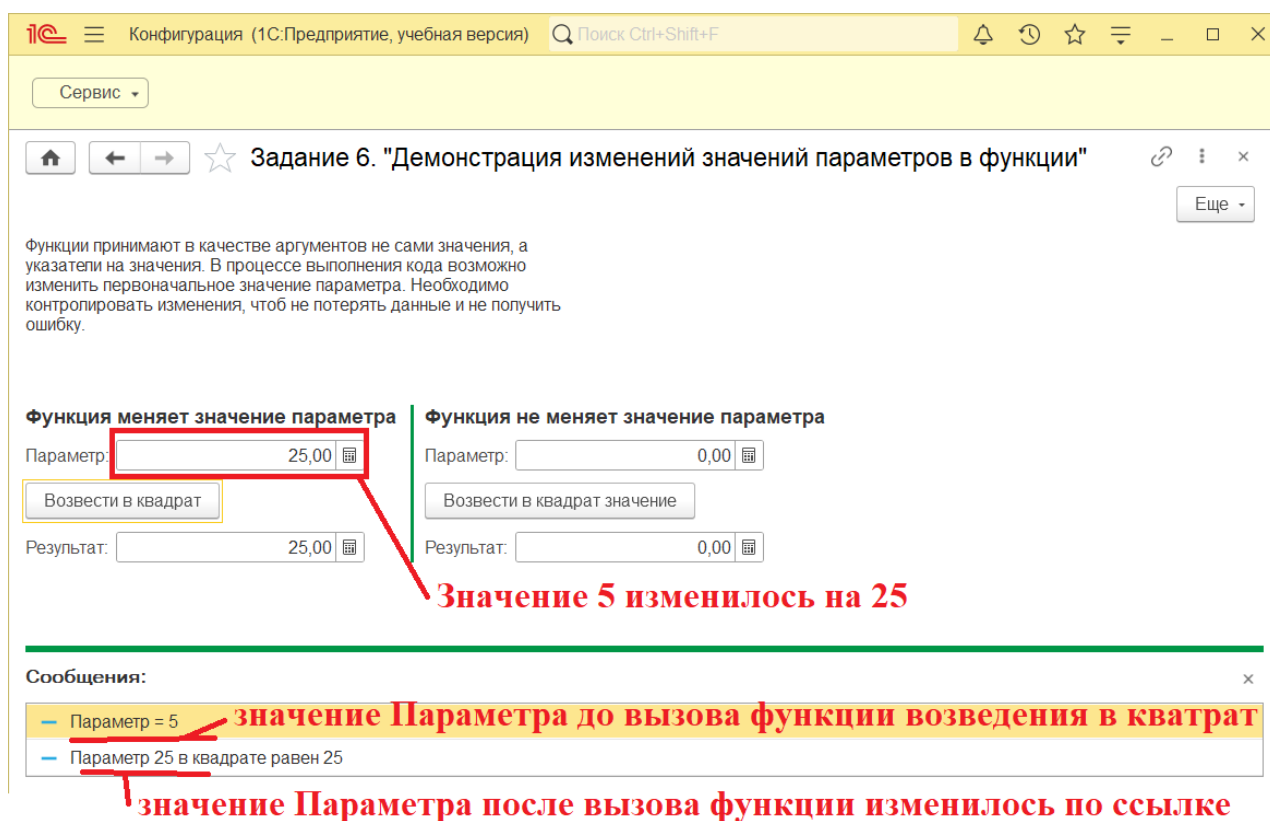
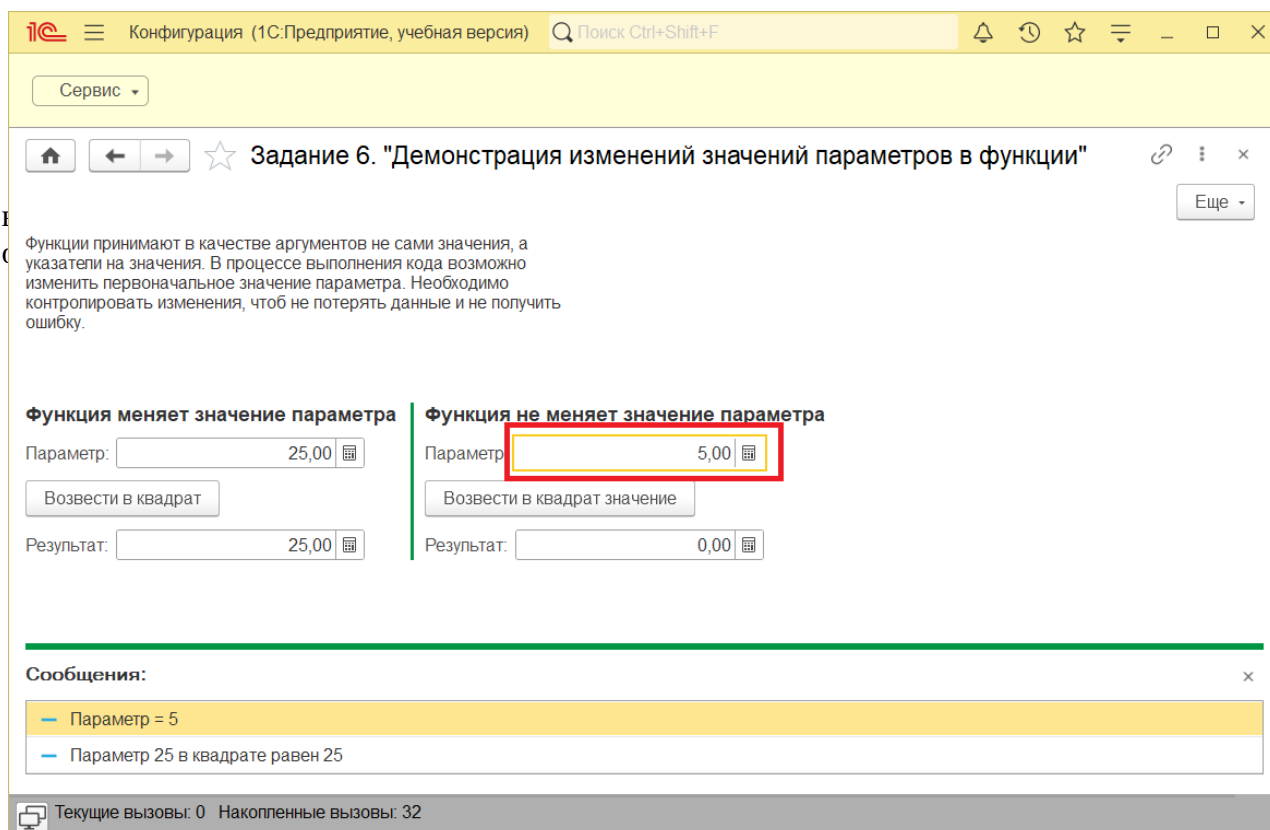


Рисунок 37 - Изменение значения Параметра по ссылке

Для демонстрации работы второго варианта функции определения квадрата с использованием ключевого слова "Знач", необходимо ввести значение, например, 5 в поле ввода Параметра справа (рисунок 38).





&НаКлиенте

Функция ВозведениеВКвадратЗначения(Знач Аргумент) Экспорт

Аргумент = Аргумент \* Аргумент;

Возврат Аргумент;

КонецФункции

В теле функции ВозведениеВКвадратЗначения(Знач Аргумент) такой же код, как и в теле Функции ВозведениеВКвадрат(Аргумент). Разница заключается в наличии ключевого слова "Знач". В описании функции ВозведениеВКвадратЗначения(Знач Аргумент) перед Аргументом находится ключевое слово "Знач", поэтому исходное значение параметра остается не измененным (рисунок 39).

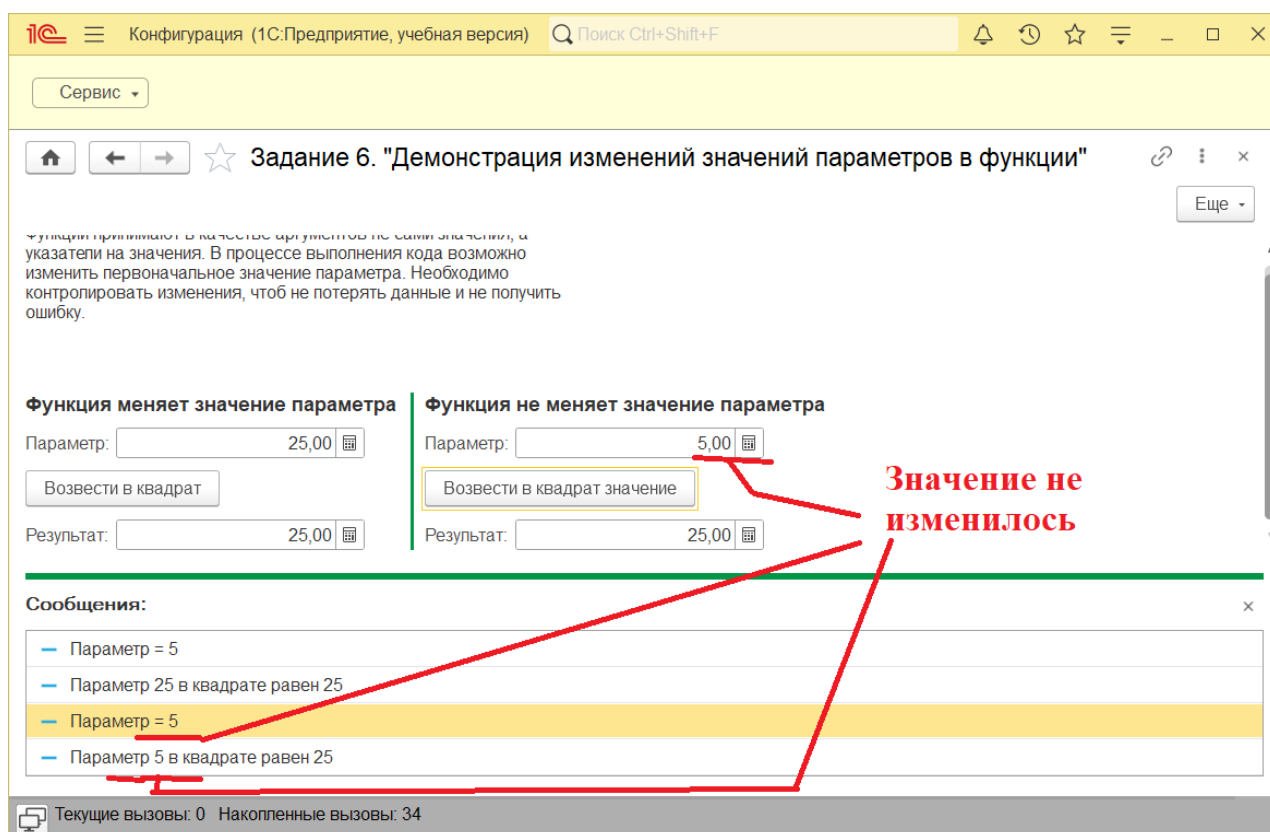


Рисунок 39 - Параметр не изменился

### Вывод по заданию.

1. При возникновении каких-либо вопросов по синтаксису или использованию операторов и конструкций лучше в первую очередь обращаться к Синтаксис-помощнику платформы 1С:Предприятие. Во вторую очередь - к поиску материалов в сети Интернет.
2. При написании функций внимательно следить, когда нам нужны только значения параметров, а когда можно использовать ссылку, таким образом чтобы не привести к потере важных данных [3].
3. Если какие-то действия повторяются в разных модулях, то использовать Экспорт из общего модуля [4].

## Список литературы

1. Синтаксис-помощник учебной платформы 1С:Предприятие.
2. "Обработка исключений" Загл. с экрана. Url: <https://its.1c.ru/db/metod8dev/content/2333/hdoc>
3. "1С 8.3 Процедуры и Функции" Загл. с экрана. Url: [https://koder.by/procedure\\_i\\_funkcii.php](https://koder.by/procedure_i_funkcii.php)
4. "Внутренний язык программирования 1С 8.3 для начинающих программистов: функции и процедуры в 1С часть". Загл. с экрана. Url: <https://helpmels.ru/vnutrennij-yazyk-programmirovaniya-1s-8-3-dlya-nachinayushhix-programmistov-urok-13>