

Лабораторная работа № 7

**ВСТРОЕННЫЙ ЯЗЫК 1С.
ПРИМИТИВНЫЕ ТИПЫ
ДАННЫХ.**



1. Общее описание встроенного языка

Встроенный язык является важной частью технологической платформы «1С:Предприятия 8», поскольку позволяет разработчику описывать собственные алгоритмы функционирования прикладного решения.

Встроенный язык имеет много общих черт с другими языками, такими как Pascal, Java Script, Basic, что облегчает его освоение начинающими разработчиками. Однако он не является прямым аналогом какого-либо из перечисленных языков.

Вот лишь некоторые, наиболее значимые особенности встроенного языка:

- Предварительная компиляция — перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код;
- Кэширование скомпилированных модулей в памяти;
- Мягкая типизация — тип переменной определяется типом значения, которое она содержит, и может изменяться в процессе работы;
- Встроенный язык 1С является предметно-ориентированным языком. Обладает некоторыми возможностями объектно-ориентированных языков: обращение к свойствам и методам объектов. Но программно нельзя создавать новые классы, только визуально.
- Динамическая типизация переменных — тип переменной определяется в процессе выполнения программы. Одна переменная может принимать значения разных типов.

В частности, все операторы языка имеют как русское, так и англоязычное написание, которое можно использовать одновременно в одном исходном тексте.

Например:

```
Message("Hello, World!");  
Сообщить("Привет, Мир!");
```

2. Комментарии

В тексте модуля все что идет после двойного слеша «//» является комментарием. При компиляции программного модуля комментарии игнорируются.

Комментарии нужны чтобы сделать код более понятным для программиста. Как правило на уровне процедуры или функции комментарии отвечают на вопрос «Что?» (что делает данная функция). Внутри процедуры на вопрос «Как?» (алгоритм функции). На уровне строки кода на вопрос «Почему?» (почему переменной было присвоено такое-то значение и т.п.)

```
//это комментарий  
a = 2; // это тоже комментарий
```

3. Операторы встроенного языка

Это основной блок программного модуля. Между собой разделяются точкой с запятой «;». На одной строке можно расположить несколько операторов. Также один оператор может быть на нескольких строках (тогда в конце первой строки не нужно ставить «;»).

Операторы бывают двух видов:

- Операторы объявления переменных;
- Исполняемые операторы, которые манипулируют с переменными.

```
Перем Переменная; //это оператор объявления переменной  
Переменная = 1; // исполняемый оператор  
МояПроцедура(Переменная); // тоже исполняемый оператор
```

При написании кода на встроенном языке можно писать на русском, на английском и даже смешивать.

```
Перем Переменная;  
Переменная = Undefined;
```

Регистр не имеет значения:



```
Переменная = 1;  
переменная = 1;  
переменНАЯ = 2; //это все одна и та же переменная
```

4. Примитивные типы данных

- Булево

Основные операции: =, >, <, >=, <=, <>. Причем Ложь < Истина
И, ИЛИ, НЕ

Булевские значения. Существует только два значения *Истина* и *Ложь*, которые могут быть получены различными способами. Можно, например, использовать операции сравнения чисел или дат. В итоге будет получаться некое булевское значение, которое в дальнейшем наиболее часто используется в условных операторах и в операторах цикла.

Применяется в основном для формирования результата логического выражения

- Числа и математика в 1С

Основные функции: Цел, Окр, Log, Log10, Sin, Cos, Tan, ASin, ACos, ATan, Exp, Pow, Sqrt

Основные операции: +, −, *, /, %; =, >, <, >=, <=, <>

Числовыми могут быть переменные модулей и реквизиты объектов базы данных. Для числа существует ограничение разрядности. Для числового реквизита длина целой части не может превышать 32 символов. В качестве разделителя целой и дробной части используется "."

Точность дробной части не может превышать 10 цифр. Когда описывается переменная и присваивается ей числовое значение, то нигде не фиксируется ее разрядность. Тем не менее, для переменных тоже существуют ограничения.

Максимально допустимая разрядность для числа – это 38 знаков.



Такое ограничение не препятствует решению любых экономических задач, т.е. любую денежную величину можно описать с помощью этих чисел. Тем не менее, если все-таки потребуется описать большие величины для решения каких-то математических задач, то в теории программирования есть алгоритмы, позволяющие описать числа с любой размерностью на основании существующих ограничений.

Выполнение многочисленных, часто применяемых простых математических операций рассмотрены на конкретных нижеследующих примерах.

```
/// Как округлить число
    // до второго знака после запятой
Сообщить (Окр(321.123, 2)); // 321.12
    // до второго знака перед запятой
Сообщить (Окр(321.123, -2)); // 300
    // по умолчанию округление работает
    // в большую сторону
Сообщить (Окр(1.5, 0)); // 2
    // а если указать третьим параметром 0,
    // то в меньшую
Сообщить (Окр(1.5, 0, 0)); // 1

/// Как получить целую часть числа
Сообщить (Цел(12.654)); // 12

/// Как возвести число в степень
Сообщить (Pow(10, 3)); // 10^3 = 1000

/// Как извлечь квадратный корень из числа
Сообщить (Sqrt(25)); // 5

/// Как вычислить максимальное из
/// нескольких значений
Сообщить (Макс(1, 2, 3)); // 3
Сообщить (Макс("aaa", "bbb", "ccc")); // ccc
Сообщить (Макс(ТекущаяДата(),
ДобавитьМесяц(ТекущаяДата(), 1)));
Сообщить (Макс(Истина, Ложь)); // Истина

/// Как вычислить минимальное из
/// нескольких значений
Сообщить (Мин(1, 2, 3)); // 1
Сообщить (Мин("aaa", "bbb", "ccc")); // aaa
Сообщить (Мин(ТекущаяДата(),
ДобавитьМесяц(ТекущаяДата(), 1)));
Сообщить (Мин(Истина, Ложь)); // Ложь
```

```

/// Как вычислить выражение из строки
Сообщить (Вычислить ("1+2")); // 3
Сообщить (Вычислить ("ИСТИНА И ЛОЖЬ")); // Нет
Сообщить (Вычислить ("Sin(3.14)")); // ~0
Сообщить (Вычислить ("Pow(Sin(3.14), 2) + Pow
(Cos(3.14), 2)")); // ~1

/// Тригонометрия
// углы выражаются в радианах
Пи = 3.14;
Е = 2.718; // Число Эйлера
Сообщить (Cos(Пи / 2)); // ~0
Сообщить (ACos(0)); // ~1,57
Сообщить (Sin(Пи / 2)); // ~1
Сообщить (ASin(1)); // ~1,57
Сообщить (Sin(Пи / 2) * Sin(Пи / 2) +
Cos(Пи / 2) * Cos(Пи / 2)); // ~1
Сообщить (Sin(Пи / 3) / Cos(Пи / 3)); // 1.73
Сообщить (Tan(Пи / 3)); // 1.73
Сообщить (Tan(ATan(1.73))); // 1.73
// экспонента - возведение числа Эйлера
// в степень X
Сообщить (Pow(Е, 2)); // 7,389
Сообщить (Exp(2)); // 7,389
// натуральный логарифм - степень в которую надо
// возвести число Эйлера e, чтобы получить X
Сообщить (Log(9)); // 2,197
Сообщить (Pow(Е, 2.197)); // ~9
// десятичный логарифм - степень в которую надо
// возвести число 10, чтобы получить X
Сообщить (Log10(1000)); // 3
Сообщить (Pow(10, 3)); // 1000

/// Как вывести число без пробелов
Сообщить (Формат(1000000, "ЧГ=0")); // 1000000

```

• Дата

Основные функции: Год, Месяц, День, Час, Минута, Секунда, НачалоГода, НачалоМесяца, НачалоКвартала, НачалоНедели, НачалоДня, НачалоЧаса, НачалоМинуты, КонецГода, КонецМесяца, КонецКвартала, КонецНедели, КонецДня, КонецЧаса, КонецМинуты, НеделяГода, ДеньГода, ДеньНедели, ДобавитьМесяц, ТекущаяДата, Дата.

Основные операции: =, >, <, >=, <=, <>



Дата + Дата = **нельзя!!!**

Дата + Число = Дата

Дата – Дата = Число

Дата – Число = Дата

Число – количество секунд

‘20120818153854’ + 60 = ‘20120818153954’

‘20120818153854’ – 180 = ‘20120818153554’

‘20120818153854’ – ‘20120817153854’ = 86400 (= 24*60*60)

Описание:

Значения данного типа содержит дату григорианского календаря (с 01 января 0001 года) и время с точностью до 0,1 миллисекунды.

Литералы:

Строка цифр, заключенная в одинарные кавычки вида: 'ТГГГММДДччммсс', где:

- ТГГГ — четыре цифры года (включая тысячелетие и век);
- ММ — две цифры месяца;
- ДД — две цифры даты;
- чч — две цифры часа (в 24-х часовом формате);
- мм — две цифры минут;
- сс — две цифры секунд;

Во встроенном языке в литерале типа Дата обязательно должно задаваться значение года, месяца и дня. Для задания даты, соответствующей началу отсчета достаточно указать '00010101'. Допускается при указании литералов типа Дата опускать последние символы (секунды, минуты, часы и т.д.). Это означает, что данные параметры будут равны нулю (для времени) или единице (для даты). В литерале даты допускается использование различных разделителей.

/// Как инициализировать дату константой

Дата = '20130724'; // 24.07.2013

/// Как преобразовать строку в дату

```
Дата = Дата("20130724"); // 24.07.2013
/// Как инициализировать дату отдельными компонентами
Дата = Дата(2013, 07, 24); // 24.07.2013
/// Как указать в дате часы, минуты и секунды
Дата = '20130724132506'; // 24 июля 2013 г. 13 ч. 25
// мин. 6 сек.
/// Как узнать год (месяц, день, час, минута, секунда) у даты
Г = Год(Дата); // 2013
М = Месяц(Дата); // 7
Д = День(Дата); // 24
Ч = Час(Дата); // 13
Ми = Минута(Дата); // 25
С = Секунда(Дата); // 6
/// Как получить текущую дату
Сообщить(ТекущаяДата());
/// Операции с датой
Сообщить(ТекущаяДата() + 1); // прибавили секунду
Сообщить(ДобавитьМесяц(ТекущаяДата(), 1));
// прибавили месяц
Сообщить(ДобавитьМесяц(ТекущаяДата(), -1));
// отняли месяц
/// Вспомогательные функции для работы с датой
Дата = '20130110125905';
// 10 января 2013 года 12:59:05
Сообщить(ДеньГода(Дата)); // 10
Сообщить(ДеньНедели(Дата));
// 4, т.е. четверг (нумерация с понедельника)
Сообщить(НеделяГода(Дата)); // 2
Сообщить(НачалоГода(Дата)); // 01.01.2013 0:00:00
Сообщить(КонецГода(Дата)); // 31.12.2013 23:59:59
Сообщить(НачалоКвартала(Дата)); // 01.01.2013 0:00:00
Сообщить(КонецКвартала(Дата)); // 31.03.2013 23:59:59
Сообщить(НачалоМесяца(Дата)); // 01.01.2013 0:00:00
Сообщить(КонецМесяца(Дата)); // 31.01.2013 23:59:59
Сообщить(НачалоНедели(Дата)); // 07.01.2013 0:00:00
Сообщить(КонецНедели(Дата)); // 13.01.2013 23:59:59
Сообщить(НачалоДня(Дата)); // 10.01.2013 0:00:00
Сообщить(КонецДня(Дата)); // 10.01.2013 23:59:59
Сообщить(НачалоЧаса(Дата)); // 10.01.2013 12:00:00
Сообщить(КонецЧаса(Дата)); // 10.01.2013 12:59:59
Сообщить(НачалоМинуты(Дата)); // 10.01.2013 12:59:00
Сообщить(КонецМинуты(Дата)); // 10.01.2013 12:59:59
```




- **Строка**

Основные функции: Функции: СтрДлина, СокрЛ, СокрП, СокрЛП, Лев, Прав, Сред, Найти, ВРег, НРег, ТРег, Символ, КодСимвола, ПустаяСтрока, СтрЗаменить, СтрЧислоСтрок, СтрПолучитьСтроку, СтрЧислоВхождений, СтрНайти, СтрНачинаетсяНа, СтрЗаканчиваетсяНа

Основные операции: +, =, >, <, >=, <=, <>

Строка[Номер] – нельзя!!

Описание:

Значения данного типа содержат строку в формате Unicode произвольной длины.

Литералы:

Литералы строкового типа представляют собой набор символов заключенных в кавычки. Для задания в строке символа " (кавычка) необходимо записать две кавычки подряд. Кроме того, допускаются "многострочные" строковые константы. В исходном тексте многострочные константы могут задаваться двумя способами:

- Между фрагментами, представляющими отдельные строки многострочной строки, не должно встречаться никаких символов, за исключением пробелов, переводов строки и строк комментариев.
- Каждая отдельная составляющая не замыкается кавычками, а на каждой последующей строке помещен символ переноса строки | (вертикальная черта). В этом варианте комментарии допускаются, если строка начинается с символа комментария //.

```
/// Как найти длину строки
Сообщить (СтрДлина ("123456789")); // 9
```

```
/// Как взять часть строки
// как взять N символов слева
Строка = "Иван Грозный";
Сообщить (Лев (Строка, 4)); // Иван
```

```

    // как взять N символов справа
    Строка = "Иван Грозный";
    Сообщить (Прав(Строка, 7)); // Грозный
    // как взять строку из середины
    Строка = "Мама, папа, я.";
    // возьмем, начиная с 7 индекса 4 символа
    Сообщить (Сред(Строка, 7, 4)); // папа

/// Как найти подстроку
    Строка = "Привет, мир!";
    Индекс = Найти(Строка, "Привет"); // 1, строки индексируются с
единицы

/// Как узнать число вхождений подстроки в строку
    Сообщить (СтрЧислоВхождений("я, я и еще раз я.", "я")); // 3

/// Как заменить подстроку в строке
    Сообщить (СтрЗаменить("Идет сильный дождь.", "дождь", "снег"));

/// Как удалить незначащие символы из строки
    // слева
    Сообщить (СокрЛ("    из лесу елочку    ")); // "из лесу елочку    "
    // справа
    Сообщить (СокрП("    из лесу елочку    ")); // "    из лесу елочку"
    // с двух сторон
    Сообщить (СокрЛП("    из лесу елочку    ")); // "из лесу елочку"

/// Как вставить в строку перенос
    Строка =
        "Первая строка" + Символы.ПС +
        "Вторая строка" + Символы.ПС +
        "Третья строка";

/// Как работать с многострочными строками (нумерация с
/// единицы)
    Сообщить (СтрЧислоСтрок(Строка)); // 3
    Сообщить (СтрПолучитьСтроку(Строка, 2)); // "Вторая строка"

/// Верхний, нижний и титульный регистры
    Сообщить (ВРег("россия – родина моя!")); // РОССИЯ – РОДИНА МОЯ!
    Сообщить (НРег("МИР, ТРУД, МАЙ!")); // мир, труд, май
    Сообщить (ТРег("мама мыла раму")); // Мама Мыла Раму

/// Как получить код символа строки
    // код отдельного символа:
    // туда
    Сообщить (КодСимвола("А")); // 65
    // и обратно
    Сообщить (Символ(65)); // "А"

```



```
// код третьего символа в строке
Сообщить (КодСимвола ("АВС", 3)); //67
// код первого символа в строке
КодПервогоСимволаФамилии = КодСимвола ("Грозный");
Сообщить (КодПервогоСимволаФамилии);
// 1043, потому что Unicode.
```

```
/// Как преобразовать строку в число
Сообщить (Число ("125.36")); // 125,36
Сообщить (Число ("125,36")); // 125,36
```

```
/// Как преобразовать число в строку
Сообщить (25 + "10"); // 35
Сообщить (Строка (25) + "10"); // 2510
```

5. Отладка в «Платформе 1С»

Как вы знаете, выполнение программы 1С компьютером – это последовательное исполнение команд, написанных программистом. В данном юните мы научимся останавливать компьютер во время выполнения определенной команды и просматривать значения переменных в этот самый момент.

Этот процесс в 1С называется отладкой и он бывает очень полезен для того, чтобы разобраться в том, как программа работает. Благодаря отладке мы также можем легко находить ошибки, которые допустили на этапе написания кода.

Итак, давайте начнем наш эксперимент. Напишите в модуле управляемого приложения следующий код:

```
А = 3;
В = 5;
Сумма = А + В;
Сообщить (Сумма);
Сумма = Сумма * 2;
Сообщить (Сумма);
```

Код очень простой. Сначала мы определяем две переменные с именами А и В и значениями 3 и 5 соответственно. Затем складываем эти числа и кладем результат сложения в переменную с именем Сумма. Затем выводим это значение на экран. Затем удваиваем это значение. И снова выводим. Как уже было сказано, компьютер выполняет все эти команды последовательно,

одну за другой. Давайте заставим его остановиться на пятой строчке, в которой мы выводим результат сложения двух чисел. Для этого установим курсор мыши в пятую строчку:

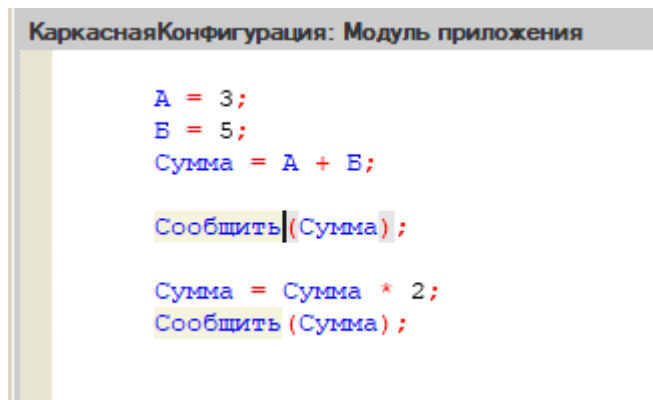


Рисунок 1 – Выбор строки для установки точки останова

И в главном меню выберем команду "Отладка"- "Точка останова" (или F9).

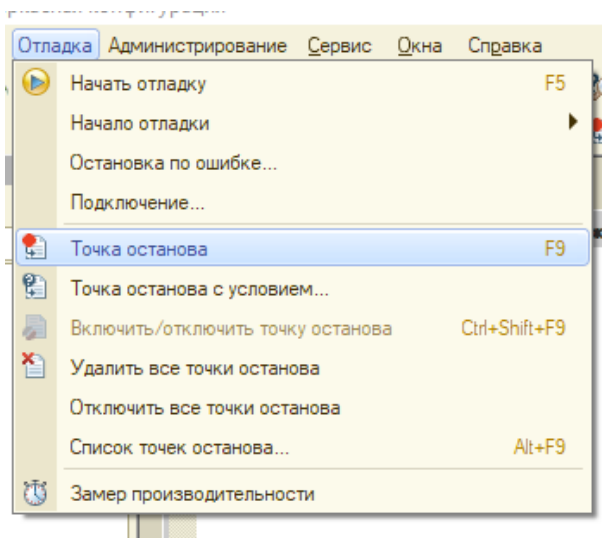


Рисунок 2 – Установка точки останова

Тогда слева от пятой строки появится красный кружок. Этот кружок и означает, что компьютер при переходе к этой строке сделает остановку. В момент остановки у нас будет возможность более детально исследовать состояние нашей программы.

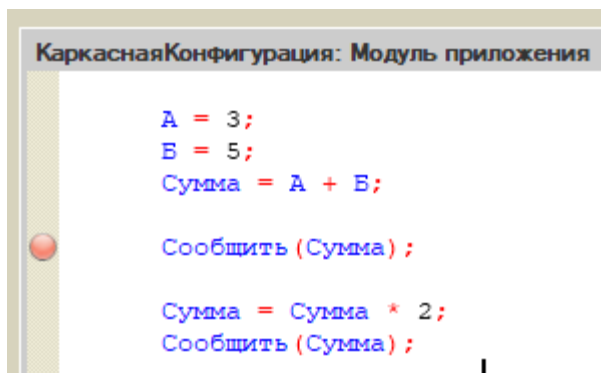


Рисунок 3 – Точка останова установлена

Теперь запустим отладку программы. Для этого через главное меню выполним команду "Отладка"- "Начать отладку" (или F5).

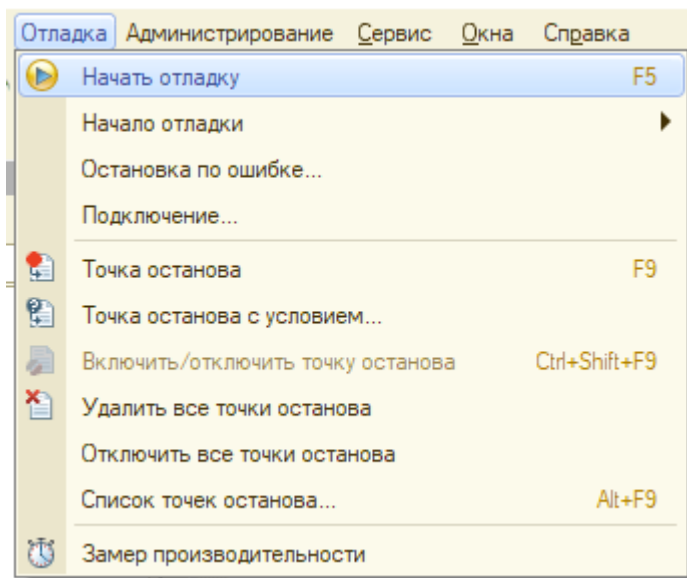


Рисунок 4 – Начало отладки

И увидим, что компьютер дошел до пятой строчки и замер в ожидании наших дальнейших действий. Об этом свидетельствует стрелка, появившаяся слева от пятой строки. Эта стрелка указывает нам, на какой команде находится компьютер в данный момент исполнения программы.

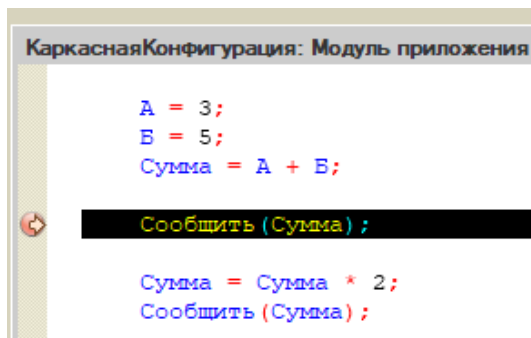


Рисунок 5 – Программа дошла до точки останова

Предположим, что мы хотим увидеть, чему равно значение переменной Сумма в данный момент. Для этого выделим мышкой эту переменную и выполним через главное меню команду "Отладка" - "Вычислить выражение...":

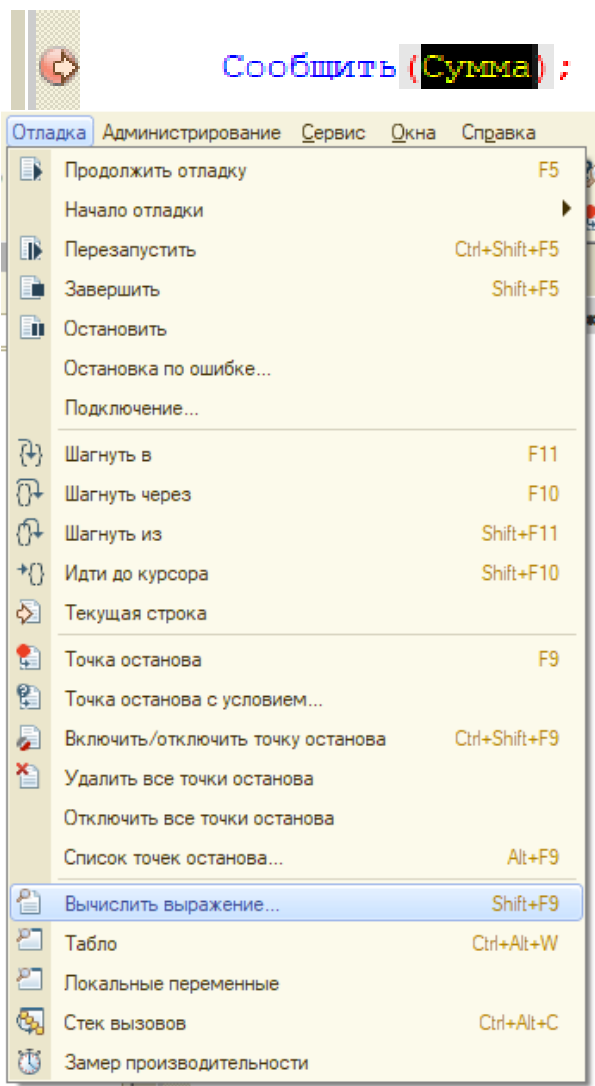


Рисунок 6 – Вычисление значения переменной в отладке

Откроется следующее окно, в котором мы видим нашу переменную и ее значение в данный момент. Оно действительно равно восьми. Закройте это окно.

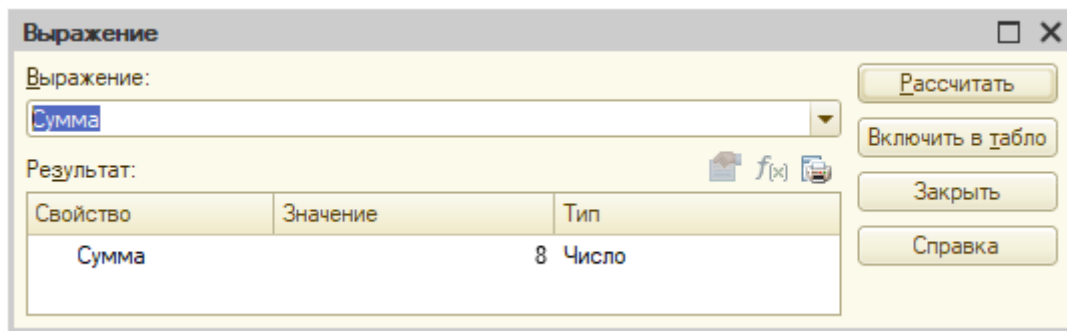


Рисунок 7 – Значение вычисленной переменной

Но давайте заставим компьютер выполнить текущую команду и перейти к следующей. Для этого выполним "Отладка"->"Шагнуть через". И мы увидим, что стрелка выполнения переместилась на следующую команду, которая удваивает значение переменной Сумма:

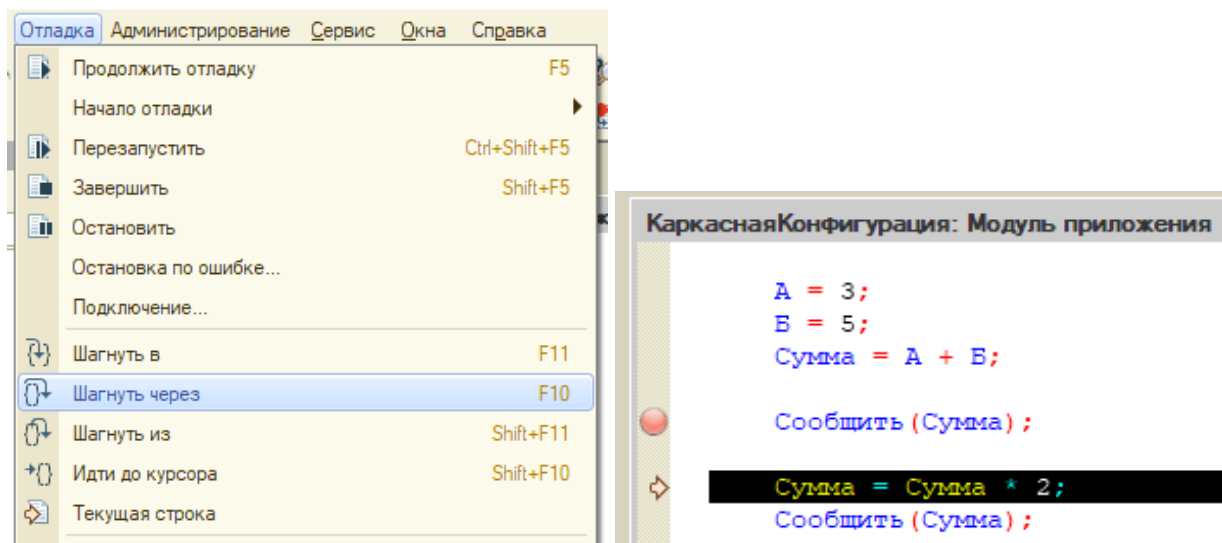


Рисунок 8 – Переход к следующей команде отладки

И еще раз выполним "Отладка"- "Шагнуть через", чтобы стрелочка оказалась на последней строке:

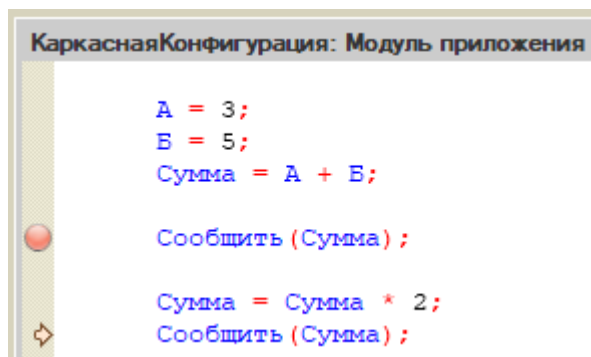


Рисунок 9 – Переход к последней команде отладки

А теперь снова выделим переменную Сумма и выполним "Отладка"— "Вычислить выражение...". Как видите, значение переменной Сумма теперь стало равно 16:

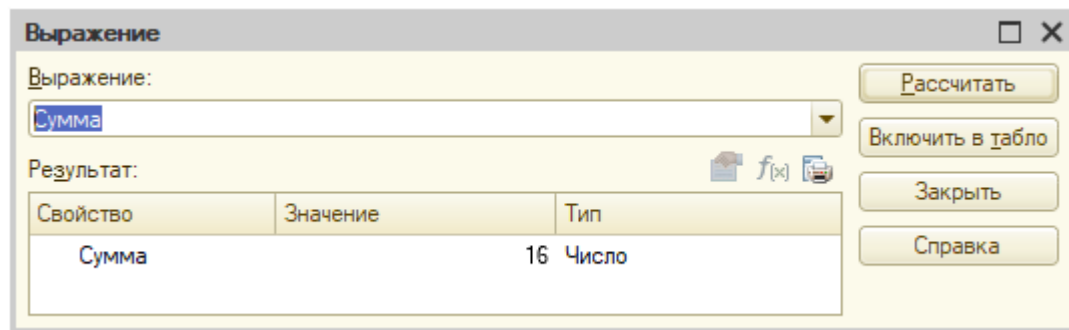


Рисунок 10 – Вычисление нового значения суммы

Чтобы прекратить отладку программы, выполним команду "Отладка"—"Завершить".

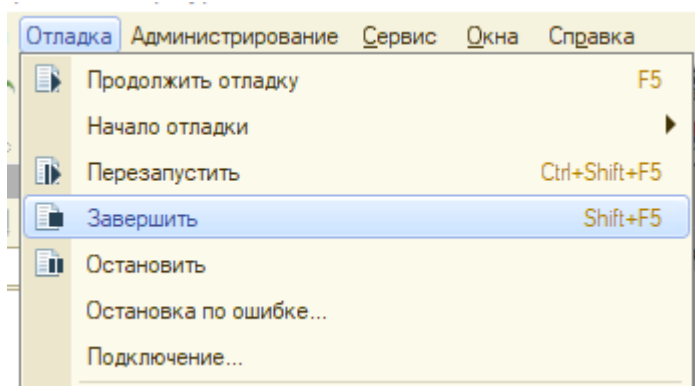


Рисунок 11 – Завершение процесса отладки

Когда отладка закончена, и мы не хотим, чтобы компьютер останавливался во время выполнения программы, просто выполним "Отладка"—"Убрать все точки останова", и все красные точки из программы исчезнут:

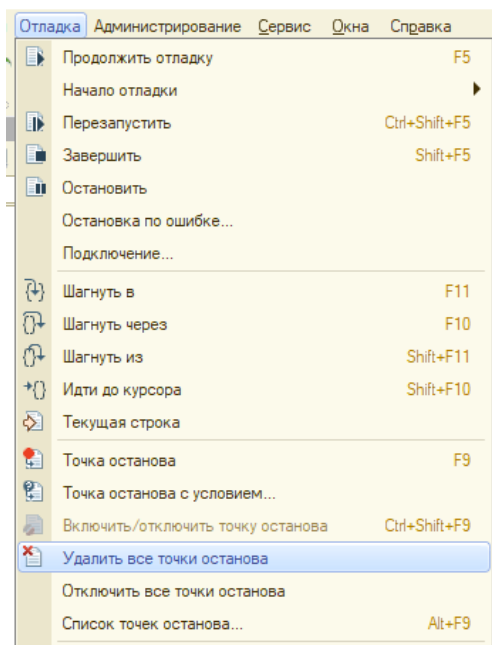


Рисунок 12 – Удаление всех точек останова

Задание на лабораторную работу

Выполнить 2 задания по варианту, используя объект дерева конфигурации «Обработка» и ее форму для ввода/вывода требуемой информации. Выбор задания осуществляется по принципу:

Задание 1: порядковый номер студента в списке группы (можно уточнить у преподавателя).

Задание 2: порядковый номер студента в группе +10.

• Вариант 1

Дана стоимость товара в копейках в виде натурального числа. Вывести стоимость в виде N руб. M коп.

Примеры:

Входные данные - Число	Выходные данные – Строка
2345	23 руб. 45 коп.
39	39 коп.

• Вариант 2

Дана строка формата "ГГГГ-ММ-ДД", необходимо преобразовать ее в переменную типа «Дата». Считаем, что строка содержит только цифры и символ "-".

Примеры:

Входные данные - Строка	Выходные данные - Дата
2018-10-08	08.10.2018
1380-09-08	08.09.1380

• Вариант 3

Даны 2 даты. Необходимо найти, сколько времени прошло между этими датами. Ответ выразить в секундах.

Примеры:

Входные данные - Дата	Выходные данные - Строка
23.03.2022 00:00:00 24.03.2022 00:00:00	Между датами прошло 86400 секунд.
23.03.2022 10:05:00 24.03.2022 10:05:30	Между датами прошло 30 секунд.

- **Вариант 4**

Даны 2 даты. Необходимо найти, сколько времени прошло между этими датами. Ответ выразить в днях.

Примеры:

Входные данные - Дата	Выходные данные - Строка
23.03.2022 00:00:00 24.03.2022 00:00:00	Между датами прошел 1 день.
23.03.2022 00:00:00 24.03.2022 12:00:00	Между датами прошло 1,5 дня.

- **Вариант 5**

Даны 2 даты. Необходимо найти, сколько времени прошло между этими датами. Ответ выразить в часах.

Примеры:

Входные данные - Дата	Выходные данные - Строка
23.03.2022 00:00:00 24.03.2022 00:00:00	Между датами прошло 24 часа.
23.03.2022 00:00:00 24.03.2022 12:00:00	Между датами прошло 36 часов.

- **Вариант 6**

Даны 2 даты. Написать программу, которая сравнивает их и выводит на экран сообщение о результате.

Примеры:

Входные данные - Дата	Выходные данные – Строка или Булево
23.03.2022 00:00:00 23.03.2022 00:00:00	«Равны» или Истина
23.03.2022 00:00:00 24.03.2022 12:00:00	«Не равны» или Ложь

- **Вариант 7**

Даны 2 строки. Написать программу, которая сравнивает их и выводит на экран сообщение о результате.

Примеры:

Входные данные - Строка	Выходные данные – Строка или Булево
«Тест» «Тест»	«Равны» или Истина
«Привет, мир» «Тест»	«Не равны» или Ложь

- **Вариант 8**

Даны 3 строки. Первую строку привести к верхнему регистру, вторую – к нижнему, третью – к титульному. В качестве результата вывести строку, состоящую из всех трех.

Примеры:

Входные данные - Строка	Выходные данные – Строка
«Привет» «МИР» «прекрасен»	«ПРИВЕТ мир Прекрасен»

- **Вариант 9**

Дана исходная строка и 2 числа. Первое число отвечает за количество символов с начала строки, второе число с конца. Вывести на экран результирующую строку, разделив части пробелом.

Примеры:

Входные данные – Строка и Число	Выходные данные – Строка
«Привет, мир» 2, 3	«Пр мир»
«Добрый вечер» 5, 1	«Добры р»

• Вариант 10

Дана исходная строка и 2 числа. Получить подстроку, начиная с символа, равного первому числу длиной, равной второму числу, удалив из результата все пробелы.

Примеры:

Входные данные – Строка и Число	Выходные данные – Строка
«Привет, мир» 2, 3	«рив»
«Добрый вечер» 5, 7	«ыйвече»

• Вариант 11

Даны 2 строки. Определить, входит ли вторая строка в состав первой. Результат представить в формате Истина/Ложь.

Примеры:

Входные данные – Строка	Выходные данные – Булево
«Привет, мир» «мир»	Истина
«Добрый вечер» «День»	Ложь

• Вариант 12

Даны 2 строки. Определить, сколько раз входит вторая строка в состав первой. Регистр не учитывать.

Примеры:

Входные данные – Строка	Выходные данные – Строка
«Привет, мир. Миру мир.» «мир»	«Подстрока входит в состав строки 3 раза»
«Добрый вечер» «День»	«Подстрока входит в состав строки 0 раз»

• Вариант 13

Даны 3 строки. Заменить в строке 1 все вхождения подстроки 2 на подстроку 3.

Примеры:

Входные данные – Строка	Выходные данные – Строка
«1С-Рарус» «1С» «2М»	«2М-Рарус»
«Добрый вечер» «вечер» «день»	«Добрый день»

• Вариант 14

Даны 3 строки. Проверить, что исходная строка начинается со строки 2 и заканчивается строкой 3.

Примеры:

Входные данные – Строка	Выходные данные – Булево
«1С-Рарус Севастополь» «1С» «Москва»	Истина, Ложь
«Добрый вечер» «Злой» «вечер»	Ложь, Истина

• Вариант 15

Дана дата и количество дней. Вывести на экран 2 даты: одна, которая была указанное количество дней назад, а другая, которая будет через указанное количество дней.

Примеры:

Входные данные – Дата, Число	Выходные данные – Дата
23.03.2022 00:00:00 3	20.03.2022 00:00:00 26.03.2022 00:00:00

- **Вариант 16**

Дана дата и количество часов. Вывести на экран 2 даты: одна, которая была указанное количество часов назад, а другая, которая будет через указанное количество часов.

Примеры:

Входные данные – Дата, Число	Выходные данные – Дата
23.03.2022 00:00:00	22.03.2022 21:00:00
3	23.03.2022 03:00:00

- **Вариант 17**

Дана дата и количество минут. Вывести на экран 2 даты: одна, которая была указанное количество минут назад, а другая, которая будет через указанное количество минут.

Примеры:

Входные данные – Дата, Число	Выходные данные – Дата
23.03.2022 00:00:00	22.03.2022 23:00:00
60	23.03.2022 01:00:00

Содержание отчета

1. Цель работы.
2. Описание варианта задания.
3. Пошаговое описание процесса выполнения варианта задания.
4. Выводы