

Алгоритмы

- Расстояние между 2-мя точками по координатам
- Асинхронный “синхронный” вопрос
- Формирование текста для поля составного типа в запросе по списку типов
- Как определить новый элемент при открытии управляемой формы?
- Програмное получение места на диске
- Использование RegExr в 1C
- Как установить цвет авто программно
- Параллельное выполнение кода в 1C
- Разработка регламентного задания
- Проверка существования реквизита формы в 1C (и не только)
- Работа с метаданными
- Как отловить события Удаление / Добавление / Изменения регистра в 1C ?
- Удалить текущую строку таблицы значений (управляемые формы)
- Генерация псевдослучайных чисел (линейный конгруэнтный метод)

Расстояние между 2-мя точками по координатам

```
// Код на 1C
Pi = 3.1415926535897932;
rlat1 = lat1 * Pi / 180;
rlon1 = lon1 * Pi / 180;
rlat2 = lat2 * Pi / 180;
rlon2 = lon2 * Pi / 180;
EarthRadius = 6371000; // по другим данным - 6372,795
Distance = EarthRadius * 2 * ASin(Sqrt((1 - Cos(rlat1 - rlat2))/2 + Cos(rlat1) * Cos(rlat2) * (1-Cos(rlon1-
rlon2))/2));
```

Формула для расчета:

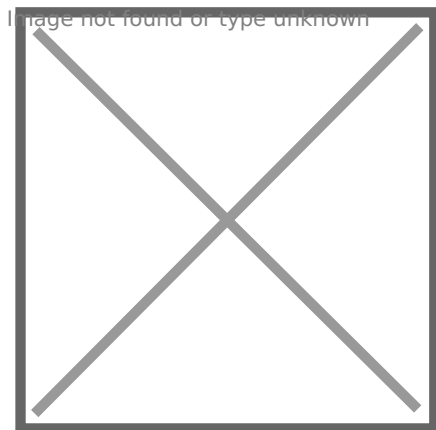
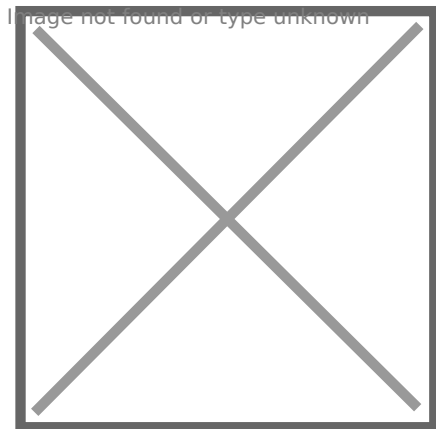


Image not found or type unknown

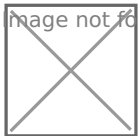


Image not found or type unknown

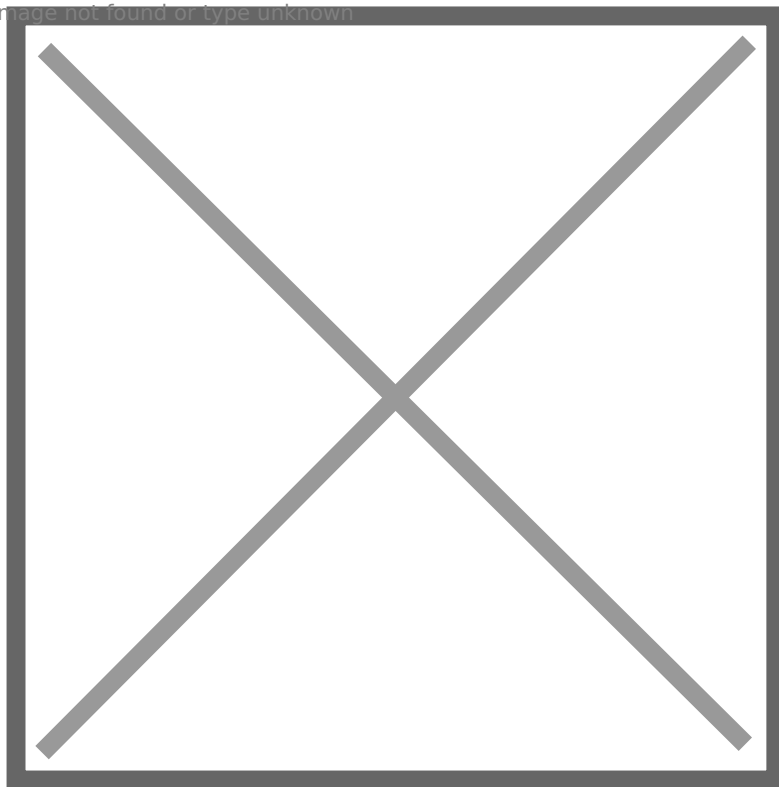
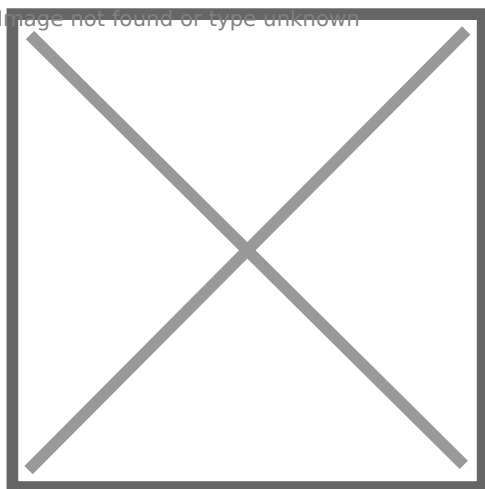
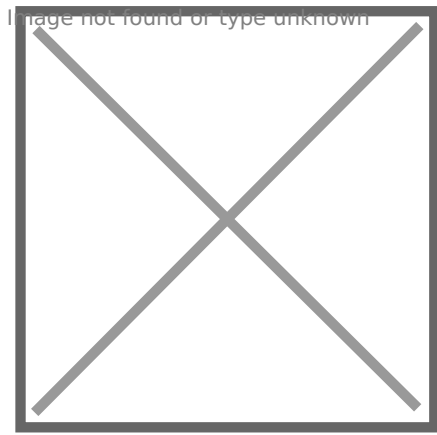


Image not found or type unknown



Радиусы земли:



Ссылки

- [Редко используемые тригонометрические функции](#)
- [Haversine_formula](#)
- [Вычисляем ближайшие объекты по координатам](#)
- [Great-circle_distance](#) (вариант на русском)
- [Расчет среднего радиуса земли](#)
- [Центральный угол](#)

P.S. Тонкости перевода: **Right angle**(правильный угол; по русски - прямой) = 90 degrees,
Straight angle(прямой угол; по-русски - развернутый) = 180 degrees

Асинхронный “синхронный” вопрос

&НаКлиенте

Асинх Процедура СинхронныйВызовНовый(Команда)

□ Ответы = Новый СписокЗначений;

□ Ответы.Добавить("Ответ1");

□ Ответы.Добавить("Ответ2");

□

□ Ответ = Ждать ВопросАсинх("Ваш ответ?", Ответы);

□ ОбщегоНазначенияКлиент.СообщитьПользователю("Ждем ответа");

□

□ Если Ответ = "Ответ2" Тогда

□ □ ОбщегоНазначенияКлиент.СообщитьПользователю("Верно");

□ Иначе

□ □ ОбщегоНазначенияКлиент.СообщитьПользователю("Не верно");

□ КонецЕсли; □

□

КонецПроцедуры

Формирование текста для поля составного типа в запросе по списку типов

В качестве оптимизации запроса при выборе в запросе поля составного типа или типа, включающего ВСЕ ссылки (например, ДокументСсылка) исходя из логики запроса можно наложить на поле ограничение по типам:

```
ВЫБОР КОГДА Регистратор ССЫЛКА Документ.ОтчетПроизводстваЗаСмену ТОГДА ВЫРАЗИТЬ(Регистратор КАК Документ.ОтчетПроизводстваЗаСмену)) .
```

Это может дать существенный прирост по производительности запроса.

Если известен список типов, который может быть выбран в запросе, то текст запроса для данного поля можно получить следующим кодом:

```
//ДоступныеТипы =
Метаданные.РегистрыБухгалтерии.Хозрасчетный.СтандартныеРеквизиты.Регистратор.Тип.Типы();
ДоступныеТипы =
РегистрыБухгалтерии.Хозрасчетный.СоздатьНаборЗаписей().Отбор.Регистратор.ТипЗначения.Типы();

ПутьКПолю      = "Регистратор";
РазделительПолей = "";
ТекстСПолям    = "";
Для Каждого Тип Из ДоступныеТипы Цикл

    ДокументСсылка = Новый(Тип);
    ИмяТипа = "Документ." + ДокументСсылка.Метаданные().Имя;
    ТекстСПолям = ТекстСПолям + РазделительПолей + СтрШаблон("ВЫБОР КОГДА %1 ССЫЛКА %2
    ||ТОГДА ВЫРАЗИТЬ(%1 КАК %2)", ПутьКПолю, ИмяТипа);
    
    РазделительПолей = Символы.ПС;
КонецЦикла;

Если ЗначениеЗаполнено(ТекстСПолям) Тогда
```

```
ТекстСПоле = СтрШаблон("%1%2КОНЕЦ", ТекстСПоле, Символы.ПС);  
КонецЕсли;
```

Как определить новый элемент при открытии управляемой формы?

```
&НаСервере  
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)[]  
[]Если Параметры.Ключ.Пустая() Тогда ...  
[]КонецЕсли;  
КонецПроцедуры
```


Програмное получение места на диске

```
fso = Новый СОМОбъект("scripting.fileSystemObject");  
Сообщить(Окр(Число(fso.getDrive(fso.GetDriveName("d:\")).freeSpace())/1024/1024, 2));
```

Использование RegExr в 1С

```
&НаКлиентеНаСервереБезКонтекста
Функция РазобратьТекст(ИсходныйТекст, Шаблон)
Попытка
    RegExr = Новый СОМОбъект(«VBScript.RegExr»);
    Иключение
    Возврат Новый Массив;
КонецПопытки;
RegExr.Multiline = Истина;
RegExr.Global = Истина;
RegExr.IgnoreCase = Истина;
RegExr.Pattern = Шаблон;
РезультатРазбора = RegExr.Execute(ИсходныйТекст);
Результат = Новый Массив;
Для каждого Item Из РезультатРазбора Цикл
    Результат.Добавить(Item.Value);
КонецЦикла;
Возврат Результат
КонецФункции
```

Как установить цвет авто программно

Источник

```
// вариант 1
ЭлементыФормы.Номенклатура.ЦветФонаПоля = ЦветаСтиля.ЦветФонаПоля;

// вариант 2
ЭлементыФормы.Номенклатура.ЦветФонаПоля = Новый Цвет();
```

Параллельное выполнение кода в 1С

&НаСервере

Процедура ВыполнитьПаралельноНаСервере()

□МассивЗаданий = Новый Массив;

□Для Каждого ТекСтрока ИЗ ТЗСклады Цикл

□□НаборПараметров = Новый Массив;

□□НаборПараметров.Добавить("%ЗначениеПараметра1%");

□□НаборПараметров.Добавить("%ЗначениеПараметраN%");

□□// запуск фонового задания

□□Задание = ФоновыеЗадания.Выполнить("%ОбщийСерверныйМодуль%.%Имяпроцедуры%",
НаборПараметров);

□□// добавляем задание в массив, что бы потом отследить выполнение

□□МассивЗаданий.Добавить(Задание);

□□

□КонецЦикла;

□□// проверим результат выполнения фоновых заданий□

□Если МассивЗаданий.Количество() > 0 Тогда

□□Попытка

□□□ФоновыеЗадания.ОжидатьЗавершения(МассивЗаданий);

□□Исключение

□□□// действия в случае ошибки

□□КонецПопытки;

□КонецЕсли;

□

КонецПроцедуры // ВыполнитьПаралельно()

Разработка регламентного задания

Перед началом выполнения регламентного задания, выполняющего интеграцию с внешними ресурсами, необходимо проверить возможность работы базы с внешними ресурсами.

```
ИмяСобытияЖР = КонстантыМодуля("ИмяСобытияЖР");
ТекстСообщенияБлокировкаРаботыСвнешнимиРесурсами =
КонстантыМодуля("ТекстСообщенияБлокировкаРаботыСвнешнимиРесурсами");
ТекстСообщенияНеВыполнена = КонстантыМодуля("ТекстСообщенияНеВыполнена");

Если БлокировкаРаботыСВнешнимиРесурсами.РаботаСВнешнимиРесурсамиЗаблокирована() Тогда
    ОписаниеОшибки = ТекстСообщенияНеВыполнена + " " +
ТекстСообщенияБлокировкаРаботыСвнешнимиРесурсами;
    #ЕСЛИ НаКлиенте ТОГДА
        Сообщить(ОписаниеОшибки);
    #КОНЕЦЕСЛИ
    ЗаписьЖурналаРегистрации(ИмяСобытияЖР, УровеньЖурналаРегистрации.Информация,,,
ОписаниеОшибки);
    
    Возврат;
КонецЕсли;

Функция КонстантыМодуля(ИмяКонстанты)
    Если ИмяКонстанты = "ИмяСобытияЖР" Тогда
        Возврат %ИмяСобытияЖР%
    ИначеЕсли ИмяКонстанты = "ТекстСообщенияБлокировкаРаботыСвнешнимиРесурсами" Тогда
        Возврат "Работа с внешними ресурсами заблокирована."
    ИначеЕсли ИмяКонстанты = "ТекстСообщенияНеВыполнена" Тогда
        Возврат "Регламентное задание прервано."
    КонецЕсли;
    
    ВызватьИсключение СтрШаблон("Константа %1 не определена", ИмяКонстанты);
    
    КонецФункции // КонстантыМодуля()
```

Проверка существования реквизита формы в 1С (и не только)

```
// Проверяет наличие реквизита или свойства у произвольного объекта без обращения к метаданным.  
Функция ЕстьРеквизитИлиСвойствоОбъекта(Объект, Имяреквизита) Экспорт  
  
    Ключуникальности = Новый Уникальныйидентификатор;  
    СтруктураРеквизита = Новый Структура (ИмяРеквизита, КлючУникальности);  
    ЗаполнитьЗначенияСвойств (СтруктураРеквизита, Объект);  
  
    Возврат СтруктураРеквизита[ИмяРеквизита] <> Ключуникальности;  
  
Конецфункции
```

Пример использования

```
Результат = ЕстьРеквизитИлиСвойствоОбъекта(Форма, "ИмяИскомоегoРеквизита");
```

Работа с метаданными

Функция МенеджерОбъектаПоСсылке(Ссылка) Экспорт

```
ТипСсылки = ТипЗнч(Ссылка);
```

```
XMLИмяТипа = XMLТип(ТипСсылки).ИмяТипа;
```

```
Возврат Новый(СтрЗаменить(XMLИмяТипа, "Ref.", "Manager."));
```

КонецФункции

Функция ЭтоСсылочныйТип(имяТипаXML) Экспорт

```
фрагменты = СтрРазделить(имяТипаXML, ".", 3);
```

```
Возврат Прав(фрагменты[0], 3) = "Ref";
```

КонецФункции

Как отловить события Удаление / Добавление / Изменения регистра в 1С ?

Источник: [20](#) , [29](#)

Добавление новой записи:

НаборЗаписей.Количество() > 0 НаборЗаписей.Модифицированность() = Истина

Изменение записи:

НаборЗаписей.Количество() = 0 НаборЗаписей.Модифицированность() = Ложь
НаборЗаписей.Количество() > 0 НаборЗаписей.Модифицированность() = Истина

Удаление записи:

НаборЗаписей.Количество() = 0 НаборЗаписей.Модифицированность() = Истина

Удалить текущую строку таблицы значений (управляемые формы)

```
Таблица = <<ТаблицаЗначений>>;
```

```
ТекущаяСтрока = Элементы.<<ТаблицаЗначений>>.ТекущаяСтрока;
```

```
Таблица.Удалить(Таблица.НайтиПоИдентификатору(ТекущаяСтрока));
```

Генерация псевдослучайных чисел (линейный конгруэнтный метод)

Формула:

$$X_{n+1} = (aX_n + c) \bmod m,$$

Примеры аргументов

| Source | <i>m</i> | множитель <i>a</i> | слагаемое <i>c</i> | используемые биты |
|---|-----------------|--------------------|--------------------|--|
| Numerical Recipes [9] | 2 ³² | 1664525 | 1013904223 | |
| Borland C/C++ | 2 ³² | 22695477 | 1 | bits 30..16 in <i>rand()</i> , 30..0 in <i>lrand()</i> |
| glibc (used by GCC) [10] | 2 ³¹ | 1103515245 | 12345 | bits 30..0 |
| ANSI C: Watcom, Digital Mars, CodeWarrior, IBM VisualAge C/C++ [11] | 2 ³¹ | 1103515245 | 12345 | bits 30..16 |

| | | | | |
|--|--------------|---|---|------------------------------|
| C99, C11: Suggestion in the ISO/IEC 9899[12] | 2^{32} | 1103515245 | 12345 | bits 30..16 |
| Borland Delphi, Virtual Pascal | 2^{32} | 134775813 | 1 | bits 63..32 of (seed * L) |
| Microsoft Visual/Quick C/C++ | 2^{32} | 214013 (343FD ₁₆) | 2531011 (269EC3 ₁₆) | bits 30..16 |
| Microsoft Visual Basic (6 and earlier)[13] | 2^{24} | 1140671485 (43FD43FD ₁₆) | 12820163 (C39EC3 ₁₆) | |
| RtlUniform from Native API [14] | $2^{31} - 1$ | 2147483629 (7FFFFFFD ₁₆) | 2147483587 (7FFFFFFC ₁₆) | |
| Apple CarbonLib, C++11's <code>minstd_rand0</code> [15] | $2^{31} - 1$ | 16807 | 0 | see MINSTD |
| C++11's <code>minstd_rand</code> [15] | $2^{31} - 1$ | 48271 | 0 | see MINSTD |
| MMIX by Donald Knuth | 2^{64} | 636413622384679 3005 | 144269504088896 3407 | |
| Newlib | 2^{64} | 636413622384679 3005 | 1 | bits 63...32 |
| VAX's MTH\$RANDOM , [16] old versions of glibc | 2^{32} | 69069 | 1 | |
| Java | 2^{48} | 25214903917 | 11 | bits 47...16 |
| Ранее во многих компиляторах: | | | | |
| RANDU | 2^{31} | 65539 | 0 | |

Ссылки

<https://habr.com/ru/post/132217/>

<https://www.youtube.com/embed/-qLMTB61ZSs>

Пример кода с проверкой цикличности неповторяемого блока:

```
#include <stdio.h>

unsigned long int seed = 1;
unsigned int lcg() {
    int a = 433494438;
    unsigned int m = 99999989;
    seed = seed * a % m;

    return seed;
}

int main() {
    unsigned int r = lcg();
    unsigned int n = 1;
    while (lcg () != 433494438) {
        n++;
    }
    printf("%d\n", n);

    return 0;
}
```