

Описание макроязыка сценариев «ОРИОН-Scripts»

Оглавление

1. [На заметку.](#)
2. [Краткое описание операторов, ключевых слов и основных функций языка.](#)
 - 2.1 [Краткий перечень операторов, ключевых слов и основных функций языка.](#)
 - 2.2 [Описание операторов, ключевых слов и основных функций языка.](#)
3. [Описание объектов системы.](#)
 - 3.1 [Краткий перечень объектов системы.](#)
 - 3.2 [Описание объектов системы.](#)
 - 3.3 [Дерево объектов.](#)
4. [Шаблонные сценарии.](#)
 - 4.1 [Узел «Прибор»](#)
 - 4.2 [Узел «Считыватель»](#)
 - 4.3 [Узел «Шлейф»](#)
 - 4.4 [Узел «Камера»](#)
 - 4.5 [Узел «Выход»](#)
 - 4.6 [Узел «Группа разделов»](#)
 - 4.7 [Узел «Раздел»](#)
 - 4.8 [Узел «Ядро»](#)
 - 4.9 [Узел «Сценарий»](#)
 - 4.10 [Узел «Монитор»](#)
 - 4.11 [Узел «Оболочка»](#)
 - 4.12 [Узел «Пенал»](#)
 - 4.13 [Узел «Сотрудники»](#)
5. [Примеры сценариев.](#)
 - 5.1 [Перечень примеров сценариев.](#)
 - 5.2 [Тексты сценариев.](#)

1. [На заметку.](#)

Существует возможность посмотреть текст сценариев, созданных на основе шаблонов. Для этого требуется запустить Администратор Базы данных, выбрать пункт меню «Настройка»\«Настройки» и отметить параметр «Показывать текст сценария в режиме ввода шаблонов».

Результат работы, предупреждения и ошибки сценария можно посмотреть в Ядре на вкладке «Сценарии».

Протестировать наличие явных ошибок и увидеть предупреждения по коду сценария, можно нажав кнопку «Проверка» в режиме редактирования сценария.

Интерпретатор скриптового языка не претендует на статус профессионального, поэтому проверку корректности приведения типов переменных, проверку обращения к еще не созданному объекту, проверку цикла на бесконечную работу и т.п. не производится. Рекомендуется внимательней отнестись к написанию операторов присваивания.

Внимание! Сценарий выполняется до первой критической ошибки. Требуется внимательней отнестись к написанию текста сценария.

2. [Краткое описание операторов, ключевых слов и основных функций языка.](#)

2.1. [Краткий перечень операторов, ключевых слов и основных функций языка.](#)

Сценарий	ключевое слово;
НачалоСценария	признак начала сценария;
КонецСценария	признак окончания сценария;
Переменные	ключевое слово объявления переменных;
Перем	объявление переменной;
=	оператор присваивания;
СоздатьОбъект	функция, создающая объект;
Задержка	оператор паузы;
Сообщить	оператор вывода текстового сообщения;
Выполнить	оператор выполнения сценария, хранимого в текстовом файле;
Если	оператор условного выбора If;
Тогда	ключевое слово оператора If;
Иначе	ключевое слово оператора If;
КонецЕсли	ключевое слово оператора If;
Для	оператор цикла For;
По	ключевое слово оператора цикла For;
Пока	оператор цикла While;
Цикл	ключевое слово операторов цикла For и While;
КонецЦикла	ключевое слово операторов цикла For и While;
Продолжить	оператор продолжения цикла Return;
Прервать	оператор прекращения цикла Break;
Возврат	оператор выхода из процедуры выполнения сценария;
ПустоеЗначение	логическая функция, проверяющая объект на существование;
ТипЗначения	функция, возвращающая тип значения;
Число	функция приведения типов к числу;
Строка	функция приведения типов к строке;
ДлинаСтроки	функция, возвращающая длину строки.
Лев	функция, возвращающая левую часть строки заданной длины;
Прав	функция, возвращающая правую часть строки заданной длины;
Сред	функция, вырезающая из строки указанное количество символов;
ТекущаяДата	функция, возвращающая текущую дату в виде строки;
ТекущееВремя	функция, возвращающая текущее время в виде строки;
ТекущееВремяДатаЧисло	функция, возвращающая текущие время и дату в виде числа;

ПолучитьГод	функция, возвращающая год, извлеченный из числа;
ПолучитьМесяц	функция, возвращающая месяц, извлеченный из числа;
ПолучитьДень	функция, возвращающая день, извлеченный из числа;
ПолучитьЧасы	функция, возвращающая час, извлеченный из числа;
ПолучитьМинуты	функция, возвращающая минуту, извлеченную из числа;
ПолучитьСекунды	функция, возвращающая секунду, извлеченную из числа;
ПолучитьМСекунды	функция, возвращающая миллисекунду, извлеченную из числа;
СобратьДатуИВремя	функция, возвращающая собранные дату и время в виде числа;
СобратьВремя	функция, возвращающая собранное время;
ДатаВремяВСтроку	функция преобразования даты и времени в строку;
ИзвлечьДиректорию	функция, извлекающая директорию из строки;
ИзвлечьФайл	функция, извлекающая имя файла с расширением из строки;
ПолучитьДиректорию	функция, возвращающая директорию из которой запущено Ядро;
СлучайноеЧисло	функция, возвращающая случайное число;
И	логическое И (And);
Или	логическое ИЛИ (Or);
Не	логическое НЕ (Not);
В	логическое вхождение в множество;
Попытка	оператор исключения Try;
Исключение	ключевое слово оператора исключения Try;
КонецПопытки	ключевое слово оператора исключения Try;
ОписаниеОшибки	функция возвращающая описание ошибки Try;
ВызватьИсключение	оператор вызова исключения;
Глобальная Функция НачалоФункции КонецФункции Далее ?	зарезервировано; зарезервировано; зарезервировано; зарезервировано; зарезервировано; зарезервировано;
ПолучитьСобытие	функция, возвращающая событие, по которому был запущен сценарий.
Символ	преобразование из числа в символ
Логическое	преобразование числа в логическую переменную.

Если строка начинается с символов **//**, то данная строка считается комментарием.

2.2. [Описание операторов, ключевых слов и основных функций языка.](#)

Оператор	Описание	Пример
Сценарий	Ключевое слово, с него начинается текст сценария. Синтаксис: Сценарий	Сценарий // Описания переменных НачалоСценария // Операторы КонецСценария
НачалоСценария	Признак начала сценария, с него начинается тело сценария. Синтаксис: НачалоСценария	
КонецСценария	Признак окончания сценария, им оканчивается сценарий. Синтаксис: КонецСценария	
Переменные	Ключевое слово объявления переменных. Синтаксис: Переменные	Сценарий Переменные Перем Раздел1; Перем Реле1; НачалоСценария // Операторы
Перем	Объявление переменной.	

	Синтаксис: Перем <i>ИмяПеременной</i> (*1)	КонецСценария
=	Оператор присваивания Синтаксис: <i>ИмяПеременной</i> = <i>Значение</i> ; (*2)	Счетчик1 = 1; Текст1 = "Какой-то текст";
СоздатьОбъект	Функция создающая объект. Возвращает объект. Синтаксис: <i>ИмяПеременной</i> = СоздатьОбъект("ИмяОбъекта"); (*3)	Разделы1 = СоздатьОбъект("Разделы");
Задержка	Оператор паузы. Синтаксис: Задержка(<i>ЧислоМиллисекунд</i>); (*4)	//Задержка на 2 секунды Задержка(2000);
Сообщить	Оператор вывода текстового сообщения в журнал сценариев Ядра. Синтаксис: Сообщить(<i>ТекстовоеЗначение</i> , <i>НомерСтиля</i>); (*5)	Сообщить("Внимание", 1); Сообщить(Строка(123), 0);
Выполнить	Оператор выполнения сценария, хранимого в текстовом файле. Синтаксис: Выполнить(<i>ТекстовоеЗначение</i>); (*6)	Выполнить("c:\Scr\Script.txt"); Строка1 = "c:\Script\Script 01.txt"; Выполнить(Строка1);
Если	Оператор условного выбора If. Синтаксис 1: Если <i>Условие</i> Тогда [Операторы] КонецЕсли;	Если Счетчик1 == 2 Тогда Реле1.Включить("0"); Иначе Реле1.Выключить("0"); Счетчик1 = 3; КонецЕсли;
Тогда		
Иначе	Синтаксис 2: Если <i>Условие</i> Тогда [Операторы] Иначе [Операторы] КонецЕсли; (*7)	
КонецЕсли		
Для	Операторы циклов For и While. Синтаксис цикла For: Для <i>ИмяПеременной</i> = <i>Значение1</i> По <i>Значение2</i> Цикл [Операторы] КонецЦикла;	Для Счетчик1 = 1 По 3 Цикл Разделы.Следующий(); Сообщить("Next", 0); КонецЦикла; Перемен1 = 1; Пока Перемен1 <= 3 Цикл Перемен1 = Перемен1 + 1; КонецЦикла;
По		
Пока		
Цикл		
КонецЦикла	Синтаксис цикла While: Пока <i>Условие</i> Цикл [Операторы] КонецЦикла; (*8)	
Продолжить	Оператор продолжения цикла Return. Прерывает шаг цикла и переходит на начало цикла к следующему шагу (Для цикла For счетчик цикла увеличивается на 1). Синтаксис: Продолжить;	// Снятие первых 23-х разделов // кроме 9-го Разделы1 = СоздатьОбъект("Разделы"); Разделы1.Первый(); Для Счетчик1 = 1 По 23 Цикл Раздел1 = Разделы1.Текущий(); Разделы1.Следующий(); Если Счетчик1 == 9 Тогда Продолжить; КонецЕсли; Раздел1.Снять(); КонецЦикла;
Прервать	Оператор прекращения цикла Break. Прерывает цикл и переходит к	// Снятие всех разделов до первого

	<p>оператору следующему за операторами цикла (после ключевого слова КонецЦикла).</p> <p>Синтаксис: Прервать;</p>	<p>// Снятого Разделы1 = СоздатьОбъект("Разделы"); Разделы1.Первый(); Для Счетчик1 = 1 По Разделы1.Количество() Цикл Раздел1 = Разделы1.Текущий(); Если Раздел1.Состояние() == "СНЯТ" Тогда Прервать; КонецЕсли; Раздел1.Снять(); Разделы1.Следующий(); КонецЦикла;</p>
Возврат	<p>Оператор выхода из процедуры выполнения сценария. Полностью прерывает выполнение сценария.</p> <p>Синтаксис: Возврат(0);</p>	<p>Возврат(0);</p>
ПустоеЗначение	<p>Логическая функция проверяющая объект на существование. Возвращает: ИСТИНА - если объект существует, ЛОЖЬ - если объект не существует.</p> <p>Синтаксис: ПустоеЗначение(<i>ИмяПеременной</i>)</p>	<p>Если ПустоеЗначение(Реле1) Тогда Сообщить("Реле не найдено", 1); КонецЕсли;</p>
ТипЗначения	<p>Функция возвращающая тип значения в текстовом виде. Возвращает строки: "СТРОКА", "ЧИСЛО", "ЛОГИКА", "<ОБЪЕКТ>", "ПУСТО".</p> <p>Синтаксис: ТипЗначения(<i>ИмяПеременной</i>)</p>	<p>Сообщить(ТипЗначения(Разделы1), 0);</p> <p>Число1 = 123; Текст1 = ТипЗначения(Число1);</p>
Число	<p>Функция приведения типов к числу.</p> <p>Синтаксис: Число(<i>Значение</i>) (*9)</p>	<p>Число1 = Число("4576"); Число2 = 1 - Число(ЛОЖЬ);</p> <p>Логика2 = ИСТИНА И Логика1; Число3 = Число(Не Логика2);</p>
Строка	<p>Функция приведения типов к строке.</p> <p>Синтаксис: Строка(<i>Значение</i>) (*10)</p>	<p>Строка1 = Строка(529);</p> <p>Логика2 = ИСТИНА И Логика1; Строка2 = Строка(Не Логика2);</p> <p>Сообщить(Строка(Разделы1.Текущий().Индекс()), 0);</p>
ДлинаСтроки	<p>Функция возвращающая длину строки.</p> <p>Синтаксис: ДлинаСтроки(<i>ТекстовоеЗначение</i>)</p>	<p>Число1 = ДлинаСтроки("Да");</p> <p>Строка1 = Раздел1.Название(); Сообщить(Строка(ДлинаСтроки(Строка1), 2);</p>
Лев	<p>Функция возвращающая левую часть строки, заданной длинны.</p> <p>Синтаксис: Лев(<i>ТекстовоеЗначение</i> , <i>ЧислоСимволов</i>)</p>	<p>// Возвращается строка "УРА" Строка1 = Лев("УРАган", 3);</p>
Прав	<p>Функция возвращающая правую часть строки, заданной длинны.</p> <p>Синтаксис:</p>	<p>// Возвращается 10 правых символов названия раздела Число1 = 10;</p>

	Прав(ТекстовоеЗначение , ЧислоСимволов)	Строка1 = Прав(Раздел1.Название(), Число1);
Сред	Функция вырезает из строки указанное количество символов. Синтаксис: Сред(ТекстовоеЗначение , НомерПервогоСимвола , ЧислоСимволов)	// Возвращается строка "Танки" Строка1 = Сред("ОсТанкино", 3, 5);
ТекущаяДата	Функция возвращающая текущую дату в текстовом виде. Синтаксис: ТекущаяДата()	Текст1 = ТекущаяДата(); Сообщить(Текст1, 0);
ТекущееВремя	Функция возвращающая текущее время в текстовом виде. Синтаксис: ТекущееВремя()	Сообщить(ТекущееВремя() , 0);
ТекущееВремяДатаЧисло	Функция, возвращающая текущие время и дату в виде числа. Синтаксис: ТекущееВремяДатаЧисло()	ВремяДата1 = ТекущееВремяДатаЧисло();
ПолучитьГод	Функция, возвращающая год, извлеченный из числа, содержащего время и дату. Синтаксис: ПолучитьГод(ВремяДата)	Д1 = ПолучитьГод(ВремяДата1);
ПолучитьМесяц	Функция, возвращающая месяц, извлеченный из числа, содержащего время и дату. Синтаксис: ПолучитьМесяц(ВремяДата)	Д2 = ПолучитьМесяц(ВремяДата1);
ПолучитьДень	Функция, возвращающая день, извлеченный из числа, содержащего время и дату. Синтаксис: ПолучитьДень(ВремяДата)	Д3 = ПолучитьДень(ВремяДата1);
ПолучитьЧасы	Функция, возвращающая час, извлеченный из числа, содержащего время и дату. Синтаксис: ПолучитьЧасы(ВремяДата)	В1 = ПолучитьЧасы(ВремяДата1);
ПолучитьМинуты	Функция, возвращающая минуту, извлеченную из числа, содержащего время и дату. Синтаксис: ПолучитьМинуты(ВремяДата)	В2 = ПолучитьМинуты(ВремяДата1);
ПолучитьСекунды	Функция, возвращающая секунду, извлеченную из числа, содержащего время и дату. Синтаксис: ПолучитьСекунды(ВремяДата)	В3 = ПолучитьСекунды(ВремяДата1);
ПолучитьМСекунды	Функция, возвращающая миллисекунду, извлеченную из числа, содержащего время и дату. Синтаксис: ПолучитьМСекунды(ВремяДата)	В4 = ПолучитьМСекунды(ВремяДата1);
СобратьДатуИВремя	Функция, возвращающая собранные дату и время в виде числа, собранного из: года, месяца, дня, часа, минуты, секунды и миллисекунды. Синтаксис: СобратьДатуИВремя(Год , Месяц , День , Часы , Минуты , Секунды ,	Д1 = 2011; Д2 = 8; Д3 = 21; В1 = 8; В2 = 30; В3 = 0; В4 = 0; ВремяДата2 =

	Миллисекунды)	СобратьДатуИВремя(Д1 , Д2 , Д3 , В1 , В2 , В3 , В4);
СобратьВремя	Функция, возвращающая собранное время в виде числа, собранного из: часа, минуты, секунды и миллисекунды. Синтаксис: СобратьВремя(Часы , Минуты , Секунды , Миллисекунды)	Время1 = СобратьВремя(В1 , В2 , В3 , В4);
ДатаВремяВСтроку	Функция преобразования даты и времени в строку. Синтаксис: ДатаВремяВСтроку(ВремяДата)	Сообщить(ДатаВремяВСтроку(ВремяДата2) , 0);
ИзвлечьДиректорию	Функция, извлекающая директорию из строки. Синтаксис: ИзвлечьДиректорию(ПолноеИмяФайла)	А = "C:\1c\1cSetup.exe"; Б = ИзвлечьДиректорию(А);
ИзвлечьФайл	Функция, извлекающая имя файла с расширением из строки. Синтаксис: ИзвлечьФайл(ПолноеИмяФайла)	А = "C:\1c\1cSetup.exe"; Б = ИзвлечьФайл(А);
ПолучитьДиректорию	Функция, возвращающая директорию из которой запущено Ядро опроса. Синтаксис: ПолучитьДиректорию()	Б = ПолучитьДиректорию();
СлучайноеЧисло	Функция, возвращающая случайное число в диапазоне от 0 до верхнего порога. Синтаксис: СлучайноеЧисло(ВерхнийПорог)	Х = СлучайноеЧисло(100);
И	Логическое И (And). Синтаксис: Х И Y, где Х и Y - логические значения.	Логика2 = ИСТИНА И Логика1; Если Не ((Счетчик < 10) Или (Счетчик > 5)) Тогда Пока Счетчик1 В [5, 7, 9, 10] Цикл Счетчик1 = Счетчик1 + 1; // Операторы КонецЦикла; КонецЕсли;
Или	Логическое ИЛИ (Or). Синтаксис: Х Или Y, где Х и Y - логические значения.	
Не	Логическое НЕ (Not). Синтаксис: Не X, где Y - логическое значение.	
В	Логическое вхождение в множество. Синтаксис: X В Множество, где X - значение. (*11)	
Попытка	Оператор обработки исключения Try. Синтаксис: Попытка	Для Счетчик1 = -5 По 5 Цикл Попытка Сообщить(Строка(120 / Счетчик1) , 0); Исключение Сообщить("Деление на 0" , 2); КонецПопытки; КонецЦикла;
Исключение	[Операторы] Исключение	
КонецПопытки	[Операторы] КонецПопытки; (*12)	
ОписаниеОшибки	Функция возвращающая описание ошибки Try, Синтаксис: ОписаниеОшибки() (*12)	Если ОписаниеОшибки() == "Деление на ноль!" Тогда Сообщить("Div. by 0",0); КонецЕсли;
ВызватьИсключение	Оператор вызова исключения. Синтаксис:	ВызватьИсключение("Деление на 0");

	ВызватьИсключение(ТекстовоеЗначение); (*12)	
ПолучитьСобытие	Функция, возвращающая событие, по которому был запущен сценарий. Синтаксис: ПолучитьСобытие() (*13)	Событие1= ПолучитьСобытие();
Символ	Функция преобразует число в символ	(Пример вставки перевода коретки в строку) СЦЕНАРИЙ ПЕРЕМЕННЫЕ ПЕРЕМ a; ПЕРЕМ z; НАЧАЛОСЦЕНАРИЯ a = "Hello my"+ Символ(13)+ Символ(10) + " World"; Сообщить(a,1); z = СоздатьОбъект("ЯДРО"); z.ЗАПИСЬВФАЙЛ("d:\test.txt", 0, a); КОНЕЦСЦЕНАРИЯ
Логическое	преобразование числа в логическую переменную	Логическое2 = Логическое(0);//FALSE

Примечания:

***1** - Имя переменной может содержать русские буквы и цифры, причем начинаться должно обязательно с буквы.

Переменные могут быть 4-х типов:

- Целочисленная,
- Вещественная,
- Текстовая,
- Логическая (ЛОЖЬ, ИСТИНА),
- Объект.

Тип переменной не указывается, так как он целиком зависит от присваиваемого переменной значения.

Также переменная может быть пустой, то есть не хранить никакого значения (см. следующее примечание). Какое значение (пустое или нет) хранит переменная можно проверить с помощью функции ПустоеЗначение.

***2** - В качестве значения присваиваемого переменной может выступать:

- Nil (пустое значение),
- Константа,
- Переменная,
- Функция,
- Выражение.

Примеры:

```
Текст1 = Nil;
Текст1 = "Какой-то текст";
Текст2 = Текст1;
Текст3 = Раздел1.Название();
Текст4 = Текст1 + " " + Строка( Число1 );
// или
Число1 = Nil;
Число1 = 123;
Число 2 = Число1;
Число 2 = 12,5;
Число 3 = Раздел1.Индекс();
Число 4 = ( Число 1 - 235 ) / Число( "256" );
```


Проверка корректности приведения типов переменных не производится. Рекомендуется внимательней отнестись к написанию операторов присваивания.

Особенно требуется обратить внимание на то, что пустое значение можно присвоить какой-либо переменной, но нельзя использовать пустое значение в выражении. Например, оператор:

Счетчик1 = 123 + Nil;

Является некорректным и вызовет ошибку.

***3** - Объект требуется создавать функцией СоздатьОбъект, если он не создается вызовом какой-либо соответствующей функции.

Например:

```
Сценарий
Переменные
  Перемен Разделы1;
  Перемен Раздел1;
НачалоСценария
  Разделы1 = СоздатьОбъект( "Разделы" );
  Если не ПустоеЗначение( Разделы1 ) Тогда
    Раздел1 = Разделы1.ПолучитьПоИД( 1 );
  // ...
  КонецЕсли;
КонецСценария
```

Как видно из примера, для переменной Разделы1 требуется вызов функции СоздатьОбъект(), а для переменной Раздел1 не требуется, так как функция ПолучитьПоИД() возвращает объект Раздел по номеру раздела.

Внимание!

При использовании данной функции создаются коллекции соответствующих объектов того рабочего места, где запускается сценарий управления.

Исключение - объекты, не привязанные к рабочим местам: Компьютеры, ЗоныДоступа, ГруппыДоступа, Подразделения, Сотрудники, Пароли, Скрипты, Телефоны.

Еще один пример корректного и некорректного написания сценария для одной и той же задачи:

1. Пример некорректного сценария:
Разделы1 = СоздатьОбъект("Разделы");
ТекстоваяПеременная1 = Разделы1.Первый().Название();
2. Пример корректного сценария для данной задачи:
Разделы1 = СоздатьОбъект("Разделы");
Разделы1.Первый();
Раздел1 = Разделы1.Текущий();
ТекстоваяПеременная1 = Раздел1.Название();

Ошибка, описанная в примере некорректного сценария, не отслеживается АБД при проверке, поэтому рекомендуется внимательнее подойти к созданию объектов системы.

Список объектов системы, которые можно создать, приведен в пункте 3.1 настоящего руководства.

***4** - В качестве параметра указывается целое положительное число от 1 до 2147483647

***5** - В качестве **Текстового значения** может выступать:




- Текстовая константа,
- Переменная, хранящая текстовую константу,
- Функция, возвращающая в качестве результата текстовую константу,
- Выражение, имеющее в качестве результата текстовую константу.

Например:

```
Сообщить( "Какой-то текст" , 0 );
Сообщить( Текст1 , 1 );
Сообщить( Строка( 123 ) , 2 );
Сообщить( "Название раздела:" + Раздел1.Название() , 1 );
```

Номер стиля может иметь значение 0, 1 или 2. Если указать любое другое значение, то во время выполнения сценария произойдет ошибка и сценарий прервется.

Описание стилей:

- 0 - Обычное сообщение, текст сообщения выводится с пиктограммой  ;
- 1 - Предупреждение, текст сообщения выводится с пиктограммой  ;
- 2 - Ошибка, текст сообщения выводится с пиктограммой .

***6** - В Текстовом значении должно храниться полное имя файла: путь + имя файла + расширение.

В качестве Текстового значения может выступать:

- Текстовая константа,
- Переменная, хранящая текстовую константу,
- Функция, возвращающая в качестве результата текстовую константу,
- Выражение, имеющее в качестве результата текстовую константу.

Например:

```
Выполнить( "c:\Scripts\Script01.txt" );  
Выполнить( Строка1 );
```

***7** - В Условии могут использоваться:

а) условные операторы:

>
>=
<
<=
==
<>
В

б) логические операторы:

И
Или
Не

в) арифметические операторы:

+ - * / ()

г) константы

д) переменные

е) функции

Написание условия подобно описанию условия на многих языках программирования.

Обратите внимание на то, что операция сравнения == в условном операторе отличается по написанию от оператора присваивания =.

Примеры:

```
Счетчик1 > 1  
( ( Счетчик1 + Счетчик2 ) == 5 ) И ( Не ПустоеЗначение( СписокСценариев1 ) )
```

***8** - Интерпретатор скриптового языка проверку цикла на бесконечную работу не производит. В настоящее время это алгоритмически невозможно. Поэтому рекомендуется внимательнее относиться к написанию условий для работы циклов.

***9** - В качестве значения может использоваться константа, переменная, функция или выражение, имеющая тип:

- Логический

- если значение ИСТИНА, то возвращается 1,
- если значение ЛОЖЬ, то возвращается 0;

- Текстовый

- если значение является текстовым представлением целого числа, то возвращается его числовой эквивалент (например "1234" -> 1234),
- если значение является текстовым представлением дробного или вещественного числа, то возникает ошибка (например "12.1" или "3/4", то ошибка).

- если значение является любым другим текстом, то также возникает ошибка (например "Раздел" или "Внимание", то ошибка).

Если значение имело целочисленный тип, то возвращается его значение. В данном случае использование функции «Число» не требуется.

Значение типа <ОБЪЕКТ> использовать в качестве параметра функции «Число» некорректно.

***10** - В качестве значения может использоваться константа, переменная, функция или выражение, имеющая тип целочисленный или логический.

Если значение имело текстовый тип, то возвращается его значение. В данном случае использование функции «Строка» не требуется.

Значение типа <ОБЪЕКТ> использовать в качестве параметра функции «Строка» некорректно.

***11** - Множество описывается как набор констант:
[Константа1, Константа2, ..., КонстантаN]

Множество может содержать от одной константы до нескольких, но не может быть пустым. Константы, используемые при описании множества, могут быть целочисленного, логического и текстового типов.

Примеры:

```
[1, 2, 3, 5, 9]
[ЛОЖЬ, ИСТИНА]
["Комната 1", "Коридор 3", "Холл"]
[1, "Комната 1", ЛОЖЬ, 3]
```

***12** - Пример использования исключения:

```
Сценарий
  Переменные
    Перемен Счетчик1;
НачалоСценария
  Для Счетчик1 = -5 По 5 Цикл
    Попытка
      Сообщить( Строка( 120 / Счетчик1 ) , 0);
    Исключение
      Сообщить( "Деление на 0" , 2);
    КонецПопытки;
  КонецЦикла;
КонецСценария
```

Пример использования исключения с помощью оператора вызова исключения:

```
Сценарий
  Переменные
    Перемен Счетчик1;
НачалоСценария
  Для Счетчик1 = -5 По 5 Цикл
    Попытка
      Если Счетчик1 <> 0 Тогда
        Сообщить( Строка( 120 / Счетчик1 ) , 0);
      Иначе
        ВызватьИсключение( "Деление на ноль!" );
      КонецЕсли;
    Исключение
      Если ОписаниеОшибки() == "Деление на ноль!" Тогда
        Сообщить( "Деление на ноль!" , 0 );
      КонецЕсли;
    КонецПопытки;
  КонецЦикла;
КонецСценария
```

*13 - Пример анализа события, по которому произошел запуск сценария:

Сценарий

Переменные

Перем Событие1;

НачалоСценария

Событие1 = ПолучитьСобытие();

Если Не ПустоеЗначение(Событие1.Событие())

Тогда

Сообщить("Событие: " + Строка(Событие1.Событие().ИД()) + " - " +
Событие1.Событие().Название() , 0);

Иначе

Сообщить("Ошибка выполнения сценария... Выход!" , 1);

Возврат(0);

КонецЕсли;

Если Событие1.ВремяДата() <> 0 Тогда

Сообщить("Дата и время: " + ДатаВремяВСтроку(Событие1.ВремяДата()) , 0);

Иначе

Сообщить("Дата и время отсутствуют..." , 1);

КонецЕсли;

Если Не ПустоеЗначение(Событие1.Прибор())

Тогда

Сообщить("Прибор: " + Событие1.Прибор().Название() , 0);

Иначе

Сообщить("Событие не приборное..." , 0);

КонецЕсли;

Если Не ПустоеЗначение(Событие1.Элемент())

Тогда

Сообщить("Элемент - " + Событие1.Элемент().Тип() + ": " + Событие1.Элемент().Название() , 0);

Иначе

Сообщить("Событие не для зоны..." , 0);

КонецЕсли;

Если Не ПустоеЗначение(Событие1.Объект())

Тогда

Сообщить("Раздел \ Группа разделов: " + Событие1.Объект().Название() , 0);

Иначе

Сообщить("Событие не для раздела\группы..." , 0);

КонецЕсли;

Если Не ПустоеЗначение(Событие1.Пароль())

Тогда

Сообщить("Идентификатор: " + Событие1.Пароль().СтроковыйПароль() , 0);

Иначе

Сообщить("Идентификатор отсутствует..." , 0);

КонецЕсли;

Если Не ПустоеЗначение(Событие1.Дверь())

Тогда

Сообщить("Точка доступа: " + Событие1.Дверь().Название() , 0);

Если Событие1.РежимПрохода() <> -1

Тогда

Если Событие1.РежимПрохода() == 0

Тогда

Сообщить("Направление: Проход" , 0);

Иначе

```
Если Событие1.РежимПрохода() == 1
Тогда
    Сообщить( "Направление: Вход" , 0);
Иначе
    Если Событие1.РежимПрохода() == 2
    Тогда
        Сообщить( "Направление: Выход" , 0);
        КонецЕсли;
    КонецЕсли;
КонецЕсли;

Иначе
    Сообщить( "Событие не для точки доступа..." , 0 );
КонецЕсли;

Если Не ПустоеЗначение( Событие1.ЗонаДоступа() )
Тогда
    Сообщить( "ЗонаДоступа: " + Событие1.ЗонаДоступа().Название() , 0 );
Иначе
    Сообщить( "Зоны доступа нет..." , 0 );
КонецЕсли;

КонецСценария
```

3. Описание объектов системы.

3.1. Краткий перечень объектов системы.

Компьютеры	→	
Компьютер	→	
РабочиеМеста	→	
РабочееМесто	→	
РабочееМесто (Оболочка)	→	(*)
РабочееМесто (Монитор)	→	(*)
РабочееМесто (Ядро)	→	(*)
Порты	→	
Порт	→	
Приборы	→	
Прибор	→	
Шлейфы	→	
Шлейф	→	
Выходы	→	
Выход	→	
Считыватели	→	
Считыватель	→	
Видеосистемы	→	
Видеосистема	→	
Камеры	→	
Камера	→	
ГруппыРазделов	→	
ГруппаРазделов	→	
Разделы	→	
Раздел	→	
Двери	→	
Дверь	→	
ЗоныДоступа	→	
ЗонаДоступа	→	
ГруппыДоступа	→	
ГруппаДоступа	→	
Подразделения	→	
Подразделение	→	
Сотрудники	→	
Сотрудник	→	
Пароли	→	
Пароль	→	
Скрипты	→	
Скрипт	→	
ПолноеСостояние	→	
Телефоны	→	
Телефон	→	
СобытиеСкрипта	→	
Событие	→	
ПочтовыеЯщики	→	
ПочтовыйЯщик	→	

(*) Данные объекты являются разновидностью объекта РабочееМесто.

3.2. Описание объектов системы.

- - функция
- - процедура

Компьютеры

- (*)Количество() : *Число* - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому компьютеру в коллекции.
- (*)Следующий() - переходит к следующему компьютеру в коллекции.
- (*)Предыдущий() - переходит к предыдущему компьютеру в коллекции.
- (*)Последний() - переходит к последнему компьютеру в коллекции.
- (*)Текущий() : *Объект* - возвращает текущий объект *Компьютер*.
- (*)Элемент(*Индекс*) : *Объект* - возвращает объект *Компьютер* по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(*ИндексКомпьютера*) : *Объект* - возвращает объект *Компьютер* по индексу компьютера.
- (*)ПолучитьПоИД(*ИДКомпьютера*) : *Объект* - возвращает объект *Компьютер* по идентификатору компьютера.
- (*)ПолучитьПоСетевомуАдресу("*IP-адрес*") : *Объект* - возвращает объект *Компьютер* по сетевому адресу компьютера.
- (*)ПолучитьПоТипуИСетевомуАдресу("*Тип*", "*IP-адрес*") : *Объект* - возвращает объект *РабочееМесто* по типу рабочего места и сетевому адресу компьютера.
Тип = ("ЯДРО", "МОНИТОР", "ОБОЛОЧКА", "ГО", "УРВ", "СЕРВЕР БД", "АБД", "СЕРВЕР МЕНЕДЖЕР", "ВИДЕО").
- (*)ПолучитьПоТипуИИндексу("*Тип*", *ИндексКомпьютера*) : *Объект* - возвращает объект *РабочееМесто* по типу рабочего места и индексу компьютера.
Тип = ("ЯДРО", "МОНИТОР", "ОБОЛОЧКА", "ГО", "УРВ", "СЕРВЕР БД", "АБД", "СЕРВЕР МЕНЕДЖЕР", "ВИДЕО").

Компьютер

- (*)Название() : *Строка* - возвращает название компьютера.
- (*)ИД() : *Число* - возвращает ИД компьютера.
- (*)Индекс() : *Число* - возвращает индекс компьютера.
- (*)Тег() : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : *Строка* - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(*Значение*) - помещает значение любого типа в тег объекта.
- (*)СетевойАдрес() : *Строка* - возвращает сетевой адрес компьютера.
- (*)РабочееМесто() : *Объект* - возвращает коллекцию рабочих мест, запускаемых на этом компьютере.
- (*)ТипЭлемента() : *Строка* - возвращает тип, [см табл. 1](#).

(*)ТипЭлементаЧисло(): Число-тип объекта в числовом значении, [см табл. 1](#).

РабочиеМеста

- (*)Количество() : *Число* - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому рабочему месту в коллекции.
- (*)Следующий() - переходит к следующему рабочему месту в коллекции.
- (*)Предыдущий() - переходит к предыдущему рабочему месту в коллекции.
- (*)Последний() - переходит к последнему рабочему месту в коллекции.
- (*)Текущий() : *Объект* - возвращает текущий объект *РабочееМесто*.
- (*)Элемент(*Индекс*) : *Объект* - возвращает объект *РабочееМесто* по индексу объекта в коллекции.
- (*)ПолучитьПоТипуСетевогоМеста(*Тип*) : *Объект* - возвращает объект *РабочееМесто* по типу рабочего места.
Тип = ("ЯДРО", "МОНИТОР", "ОБОЛОЧКА", "ГО", "УРВ", "СЕРВЕР БД", "АБД", "СЕРВЕР МЕНЕДЖЕР", "ВИДЕО").

Или Тип =

1	для Оболочки
2	для АБД
4	для ГО
8	для УРВ
16	для Сервера БД
32	для Сервер Менеджера
64	для Видео
128	для Ядра
256	для Монитора

(*)**ПолучитьПоСетевомуПорту(СетевойПорт)** : *Объект* - возвращает объект *РабочееМесто* по типу сетевого порта рабочего места.

Сетевой порт =

8080	для Ядра,
8083	для Монитора,
8082	для Оболочки,
8094	для УРВ,
8081	для АБД.

РабочееМесто

(*)**Название()** : *Строка* - возвращает название рабочего места.

(*)**ИД()** : *Число* - возвращает ИД рабочего места. ⁽¹⁾

(*)**Индекс()** : *Число* - возвращает индекс рабочего места. ⁽²⁾

(*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.

(*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.

(*)**СетевойПорт()** : *Число* - возвращает сетевой порт рабочего места.

(*)**ТипРабочегоМеста()** : *Строка* - возвращает тип рабочего места.

(*)**Локальное()** : *Логическое* - признак того, что данное рабочее место относится к тому компьютеру, на котором выполняется сценарий управления.

(*)**ЕстьСоединение()** : *Логическое* - признак того, что с данным рабочим местом установлено соединение. ⁽³⁾

(*)**Компьютер()** : *Объект* - возвращает объект *Компьютер*, которому принадлежит данное рабочее место.

⁽¹⁾ В данной версии АРМ «Орион Про» возвращается значение 0.

⁽²⁾ Индекс для объекта «РабочееМесто» равен порядковому номеру рабочего места при загрузке рабочих мест конкретного компьютера в оперативную память.

⁽³⁾ Проверяется связь только с рабочими местами типа «Оболочка», «Ядро» и «Монитор».

РабочееМесто (Оболочка)

(*)**Название()** : *Строка* - возвращает название оболочки.

(*)**ИД()** : *Число* - возвращает ИД оболочки. ⁽¹⁾

(*)**Индекс()** : *Число* - возвращает индекс оболочки. ⁽²⁾

(*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.

(*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.

(*)**СетевойПорт()** : *Число* - возвращает сетевой порт оболочки (8082).

(*)**ТипРабочегоМеста()** : *Строка* - возвращает тип оболочки (1).

(*)**Локальное()** : *Логическое* - признак того, что данная оболочка относится к тому компьютеру, на котором выполняется сценарий управления.

(*)**ЕстьСоединение()** : *Логическое* - признак того, что с данной оболочкой установлено соединение.

(*)**Компьютер()** : *Объект* - возвращает объект *Компьютер*, которому принадлежит данная оболочка.

(*)**ВоспроизвестиФайл("ПолноеИмяФайла" , "СпецКоманда")** - воспроизведение .wav файла,

СпецКоманда =

"СТОП" - остановить воспроизведение,

"ВОСПРОИЗВЕСТИ" - воспроизвести звуковой файл однократно,

"ПОВТОРЯТЬ" - бесконечно повторять воспроизведение файла.

(*) **ЗапуститьПрограмму**("ПолноеИмяФайла" , "Параметр") - запустить внешнюю программу с параметром.

(*) **ЗаписатьВЖурналСобытий**("Текст") - занесение данных в журнал событий.

(*) **ВоспроизвестиОповещение**("Сообщение" , КоличествоПовторов , ПризнакСрочности , ПаузаПередСообщением) - воспроизвести сообщение в Модуле речевого оповещения.

где: *Сообщение* : Строка - текст сообщения,

КоличествоПовторов : Целое - количество повторов сообщения,

ПризнакСрочности : Логическое - признак, по которому происходит

Оповещение (если воспроизводились несрочные сообщения, то сообщение с этим признаком будет воспроизводиться, а если наоборот, то будет поставлено в очередь сообщений),

ПаузаПередСообщением : Целое - пауза перед сообщением (измеряется в миллисекундах).

(*) **ОтключитьЗвук**() - отключение Модуля речевого оповещения (включения режима «Без звука»).

(*) **ВключитьЗвук**() - включение Модуля речевого оповещения.

(*) **КонвертироватьСообщениеВФайл**("Сообщение" , "Файл") - конвертирование текстового сообщения в файл формата wav,

где: *Сообщение* : Строка - текст сообщения,

Файл : Строка - имя и путь к файлу.

ен выполнении его, только на указанном компьютере¹.

Почта() : **Объект**(ПочтовыйЯщик) - возвращает [коллекцию объектов-Почтовый ящик](#), добавленных на данном компьютере

⁽¹⁾ В данной версии АРМ «Орион Про» возвращается значение 0.

⁽²⁾ Индекс для объекта «РабочееМесто» равен порядковому номеру рабочего места при загрузке рабочих мест конкретного компьютера в оперативную память.

РабочееМесто (Монитор)

(*) **Название**() : *Строка* - возвращает название монитора.

(*) **ИД**() : *Число* - возвращает ИД монитора. ⁽¹⁾

(*) **Индекс**() : *Число* - возвращает индекс монитора. ⁽²⁾

(*) **Тег**() : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*) **ТипТега**() : *Строка* - возвращает тип значения тега объекта.

(*) **ПоместитьВТег**(*Значение*) - помещает значение любого типа в тег объекта.

(*) **СетевойПорт**() : *Число* - возвращает сетевой порт монитора (8083).

(*) **ТипРабочегоМеста**() : *Строка* - возвращает тип монитора (256).

(*) **Локальное**() : *Логическое* - признак того, что данный монитор относится к тому компьютеру, на котором выполняется сценарий управления.

(*) **ЕстьСоединение**() : *Логическое* - признак того, что с данным монитором установлено соединение.

(*) **Компьютер**() : *Объект* - возвращает объект *Компьютер*, которому принадлежит данный монитор.

(*) **ПоказатьСообщение**("ТекстСообщения") - показать сообщение в окне.

(*) **ЗапросОператору**("ТекстЗапроса" , *ВремяЗапроса* , *ОтветПоУмолчанию*) :

Число - возвращает ответ оператора на запрос, выводимый в окне:

6 - Да,

7 - Нет,

где: *ТекстЗапроса* - запрос оператору,

ВремяЗапроса - время ожидания ответа на запрос,

ОтветПоУмолчанию - ответ по умолчанию, по истечении времени ожидания.

(*) **ВыборОператора**("ТекстЗапроса" , *ВремяЗапроса* , *ОтветПоУмолчанию* ,

["Ответ1" [, "Ответ2" [, "Ответ3" [, ...]]]] : *Число* - возвращает выбор оператора из различных вариантов ответов, выводимых в окне, количество вариантов произвольное.

(*) **ВводЗначения**("ТекстЗапроса" , *ВремяЗапроса*) : *Строка* - возвращает ответ оператора на запрос, выводимый в окне.

При истечении времени ожидания возвращается пустая строка.

- (*) **ВоспроизвестиФайл**("ПолноеИмяФайла" , "СпецКоманда") - воспроизведение .wav файла.
СпецКоманда =
"СТОП" - остановить воспроизведение,
"ВОСПРОИЗВЕСТИ" - воспроизвести звуковой файл однократно,
"ПОВТОРЯТЬ" - бесконечно повторять воспроизведение файла.
- (*) **ЗапуститьПрограмму**("ПолноеИмяФайла" , "Параметр") - запустить внешнюю программу с параметром.
- (*) **ЗаписьВФайл**("ПолноеИмяФайла" , Смещение , "ЗаписываемаяСтрока") - запись строки в текстовый файл, начиная с позиции, смещенной на указанное число байт от начала файла.
- (*) **ЧтениеИзФайла**("ПолноеИмяФайла" , Смещение , Длина) : Строка - чтение строки указанной длины из текстового файла, начиная с позиции, смещенной на указанное число байт от начала файла.
- (*) **РазмерФайла**("ПолноеИмяФайла") : Число - возвращает размер файла.
- (*) **Пауза**(ЧислоСекунд) - пауза выполнения сценария, отображаемая в окне.
- (*) **ЗапуститьХранительЭкрана**() - запуск хранителя экрана.
- (*) **СменаДежурства**() - вызов окна «Смена дежурства».
- (*) **ОтчетЗаСмену**() - генерация отчета за прошедшую смену.
- (*) **ВоспроизвестиОповещение**("Сообщение" , КоличествоПовторов , ПризнакСрочности , ПаузаПередСообщением) - воспроизвести сообщение в Модуле речевого оповещения.
где: Сообщение : Строка - текст сообщения,
КоличествоПовторов : Целое - количество повторов сообщения,
ПризнакСрочности : Логическое - признак, по которому происходит Оповещение (если воспроизводились несрочные сообщения, то сообщение с этим признаком будет воспроизводиться, а если наоборот, то будет поставлено в очередь сообщений),
ПаузаПередСообщением : Целое - пауза перед сообщением (измеряется в миллисекундах).
- (*) **ОтключитьЗвук**() - отключение Модуля речевого оповещения (включения режима «Без звука»).
- (*) **ВключитьЗвук**() - включение Модуля речевого оповещения.
- (*) **КонвертироватьСообщениеВФайл**("Сообщение" , "Файл") - конвертирование текстового сообщения в файл формата wav,
где: Сообщение : Строка - текст сообщения,
Файл : Строка - имя и путь к файлу.

ен выполнении его, только на указанном компьютер1.

Почта() : Объект(ПочтовыйЯщик) - возвращает [коллекцию объектов-Почтовый ящик](#), добавленных на данном компьютере

⁽¹⁾ В данной версии АРМ «Орион Про» возвращается значение 0.

⁽²⁾ Индекс для объекта «РабочееМесто» равен порядковому номеру рабочего места при загрузке рабочих мест конкретного компьютера в оперативную память.

РабочееМесто (Ядро)

- (*) **Название**() : Строка - возвращает название ядра.
- (*) **ИД**() : Число - возвращает ИД ядра. ⁽¹⁾
- (*) **Индекс**() : Число - возвращает индекс ядра. ⁽²⁾
- (*) **Тег**() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*) **ТипТега**() : Строка - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег**(Значение) - помещает значение любого типа в тег объекта.
- (*) **СетевойПорт**() : Число - возвращает сетевой порт ядра (8080).
- (*) **ТипРабочегоМеста**() : Строка - возвращает тип ядра (128).
- (*) **Локальное**() : Логическое - признак того, что данное ядро относится к тому компьютеру, на котором исполняется сценарий управления.
- (*) **ЕстьСоединение**() : Логическое - признак того, что с данным ядром установлено соединение.
- (*) **Компьютер**() : Объект - возвращает объект Компьютер, которому принадлежит данное ядро.
- (*) **Порты**() : Объект - возвращает коллекцию портов, принадлежащих данному

ядру.

- (*)**Приборы()** : *Объект* - возвращает коллекцию приборов, принадлежащих данному ядру.
- (*)**Шлейфы()** : *Объект* - возвращает коллекцию шлейфов приборов, принадлежащих данному ядру..
- (*)**Выходы()** : *Объект* - возвращает коллекцию выходов приборов, принадлежащих данному ядру.
- (*)**Считыватели()** : *Объект* - возвращает коллекцию считывателей приборов, принадлежащих данному ядру.
- (*)**Видеосистемы()** : *Объект* - возвращает коллекцию видеосистем, принадлежащих данному ядру.
- (*)**Камеры()** : *Объект* - возвращает коллекцию камер, принадлежащих данному ядру.
- (*)**ГруппыРазделов()** : *Объект* - возвращает коллекцию ВСЕХ групп разделов.
- (*)**Разделы()** : *Объект* - возвращает коллекцию разделов, принадлежащих данному ядру.

- (*)**Двери()** : *Объект* - возвращает коллекцию дверей, управляемых приборами, принадлежащими данному ядру.

- (*)**ЗоныДоступа()** : *Объект* - возвращает коллекцию ВСЕХ зон доступа.

- (*)**ГруппыДоступа()** : *Объект* - возвращает коллекцию ВСЕХ групп доступа.

- (*)**Сотрудники()** : *Объект* - возвращает коллекцию ВСЕХ сотрудников.

- (*)**Пароли()** : *Объект* - возвращает коллекцию ВСЕХ паролей.

- (*)**Скрипты()** : *Объект* - возвращает коллекцию ВСЕХ сценариев.

- (*)**ПоказатьСообщение**("*ТекстСообщения*") - показать сообщение в окне.

- (*)**ЗапросОператору**("*ТекстЗапроса*" , *ВремяЗапроса* , *ОтветПоУмолчанию*) :
Число - возвращает ответ оператора на запрос, выводимый в окне:

6 - Да,

7 - Нет,

где: *ТекстЗапроса* - запрос оператору,

ВремяЗапроса - время ожидания ответа на запрос,

ОтветПоУмолчанию - ответ по умолчанию, по истечении времени ожидания.

- (*)**ВыборОператора**("*ТекстЗапроса*" , *ВремяЗапроса* , *ОтветПоУмолчанию* , ["*Ответ1*" [, "*Ответ2*" [, "*Ответ3*" [, ...]]]) : *Число* - возвращает выбор оператора из различных вариантов ответов, выводимых в окне, количество вариантов произвольное.

- (*)**ВводЗначения**("*ТекстЗапроса*" , *ВремяЗапроса*) : *Строка* - возвращает ответ оператора на запрос, выводимый в окне.

При истечении времени ожидания возвращается пустая строка.

- (*)**ВоспроизвестиФайл**("*ПолноеИмяФайла*" , "*СпецКоманда*") - воспроизведение .wav файла.

СпецКоманда =

"СТОП" - остановить воспроизведение,

"ВОСПРОИЗВЕСТИ" - воспроизвести звуковой файл однократно,

"ПОВТОРЯТЬ" - бесконечно повторять воспроизведение файла.

- (*)**ЗапуститьПрограмму**("*ПолноеИмяФайла*" , "*Параметр*") - запустить внешнюю программу с параметром.

- (*)**ЗаписьВФайл**("*ПолноеИмяФайла*" , *Смещение* , "*ЗаписываемаяСтрока*") - запись строки в текстовый файл, начиная с позиции, смещенной на указанное число байт от начала файла.

- (*)**ЧтениеИзФайла**("*ПолноеИмяФайла*" , *Смещение* , *Длина*) : *Строка* - чтение строки указанной длины из текстового файла, начиная с позиции, смещенной на указанное число байт от начала файла.

- (*)**РазмерФайла**("*ПолноеИмяФайла*") : *Число* - возвращает размер файла.

- (*)**ВслатьПисьмо**("*Smtpr*" , "*Логин*" , "*Пароль*" , "*АдресИсходящегоПочтовогоЯщика*" , "*ТемаПисьма*" , "*Письмо*" , "*АдресПолучателя*" ,
ПортSMTPСервера , *ФлагSSLШифрования*) - посылка e-mail.

где: *Smtpr* - хост компьютера, типа «smtp.yandex.ru» или «smtp.mail.ru»,

Логин - логин пользователя, зарегистрированного на почтовом сервере,

Пароль - пароль пользователя,

АдресИсходящегоПочтовогоЯщика - адрес исходящего почтового ящика,

ТемаПисьма - тема письма,

Письмо - текст письма,

АдресПолучателя - адрес получателя,
ПортSMTPСервера - порт хоста.
ФлагSSISифрования-флаг включения SSI шифрования, по умолчанию

включен

- (*) **Пауза**(*ЧислоСекунд*) - пауза выполнения сценария, отображаемая в окне.
- (*) **ВоспроизвестиОповещение**(" *Сообщение* " , *КоличествоПовторов* , *ПризнакСрочности* , *ПаузаПередСообщением*) - воспроизвести сообщение в Модуле речевого оповещения.
где: *Сообщение* : Строка - текст сообщения,
КоличествоПовторов : Целое - количество повторов сообщения,
ПризнакСрочности : Логическое - признак, по которому происходит Оповещение (если воспроизводились несрочные сообщения, то сообщение с этим признаком будет воспроизводиться, а если наоборот, то будет поставлено в очередь сообщений),
ПаузаПередСообщением : Целое - пауза перед сообщением (измеряется в миллисекундах).
- (*) **ОтключитьЗвук**() - отключение Модуля речевого оповещения (включения режима «Без звука»).
- (*) **ВключитьЗвук**() - включение Модуля речевого оповещения.
- (*) **КонвертироватьСообщениеВФайл**(" *Сообщение* " , " *Файл* ") - конвертирование текстового сообщения в файл формата wav,
где: *Сообщение* : Строка - текст сообщения,
Файл : Строка - имя и путь к файлу.

⁽¹⁾ В данной версии APM «Орион Про» возвращается значение 0.

⁽²⁾ Индекс для объекта «РабочееМесто» равен порядковому номеру рабочего места при загрузке рабочих мест конкретного компьютера в оперативную память.

ОграничитьВыполнение(Компьютер1:Объект Компьютер): Логическое, запуск сценария ограничен выполнением его, только на указанном компьютер1.

Почта() : **Объект**(ПочтовыйЯщик)- возвращает [коллекцию объектов-Почтовый ящик](#), добавленных на данном компьютере

Порты

- (*) **Количество**() : *Число* - возвращает количество элементов в коллекции.
- (*) **КоличествоЭлементов**() : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*) **Первый**() - переходит к первому порту в коллекции.
- (*) **Следующий**() - переходит к следующему порту в коллекции.
- (*) **Предыдущий**() - переходит к предыдущему порту в коллекции.
- (*) **Последний**() - переходит к последнему порту в коллекции.
- (*) **Текущий**() : *Объект* - возвращает текущий объект *Порт*.
- (*) **Элемент**(*Индекс*) : *Объект* - возвращает объект *Порт* по индексу объекта в коллекции.
- (*) **ПолучитьПоАдресу**(" *АдресПорта* ") : *Объект* - возвращает объект *Порт* по указанному адресу вида "X.0.0.0" (где X - номер порта).
- (*) **ПолучитьПоИндексу**(*ИндексПорта*) : *Объект* - возвращает объект *Порт* по индексу Com-порта.
- (*) **ПолучитьПоИД**(*ИДПорта*) : *Объект* - возвращает объект *Порт* по идентификатору Com-порта.

Порт

- (*) **Название**() : *Строка* - возвращает название порта ⁽¹⁾.
- (*) **Адрес**() : *Строка* - возвращает адрес порта.
- (*) **ИД**() : *Число* - возвращает ИД порта.
- (*) **Индекс**() : *Число* - возвращает индекс порта.
- (*) **Тег**() : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*) **ТипТега**() : *Строка* - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег**(*Значение*) - помещает значение любого типа в тег объекта.
- (*) **Приборы**() : *Объект* - возвращает коллекцию подчиненных приборов ⁽²⁾.
- (*) **ТипЭлемента**() : *Строка* - возвращает тип, [см табл. 1](#).
- (*) **ТипЭлементаЧисло**() : *Число*-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ В данной версии APM «Орион Про» возвращается пустая строка: «».

⁽²⁾ В этих случаях, возвращается коллекция приборов подключенных (привязанных) непосредственно к Com-порту. Например, в случае использования схемы [Com-порт -> Пульт C2000. -> ...приборы] возвращается коллекция приборов, содержащая один единственный прибор: пульт C2000.

Приборы

- (*) **Количество()** : *Число* - возвращает количество элементов в коллекции.
- (*) **КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*) **Первый()** - переходит к первому прибору в коллекции.
- (*) **Следующий()** - переходит к следующему прибору в коллекции.
- (*) **Предыдущий()** - переходит к предыдущему прибору в коллекции.
- (*) **Последний()** - переходит к последнему прибору в коллекции.
- (*) **Текущий()** : *Объект* - возвращает текущий объект *Прибор*.
- (*) **Элемент(Индекс)** : *Объект* - возвращает объект *Прибор* по индексу объекта в коллекции.
- (*) **ПолучитьПоАдресу("АдресПрибора")** : *Объект* - возвращает объект *Прибор* по указанному адресу вида "X.Y.Z.0" (где X - номер порта, Y - адрес пульта, Z - адрес прибора).
- (*) **ПолучитьПоИндексу(ИндексПрибора)** : *Объект* - возвращает объект *Прибор* по индексу прибора.
- (*) **ПолучитьПоИД(ИДПрибора)** : *Объект* - возвращает объект *Прибор* по идентификатору прибора.

Прибор (Для каждого типа прибора - индивидуальный объект!)

- (*) **Название()** : *Строка* - возвращает название прибора.
- (*) **Адрес()** : *Строка* - возвращает адрес прибора.
- (*) **ИД()** : *Число* - возвращает ИД прибора.
- (*) **Индекс()** : *Число* - возвращает индекс прибора.
- (*) **Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*) **ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*) **ТипЭлемента()** : *Строка* - возвращает тип объекта (для данного объекта возвращается «ПРИБОР»).
- (*) **Тип()** : *Строка* - возвращает тип прибора ⁽¹⁾.
- (*) **Версия()** : *Строка* - возвращает версию прибора.
- (*) **Состояние()** : *Строка* - возвращает основное состояние прибора ⁽²⁾.
- (*) **ЗначениеСостояния()** : *Число* - возвращает код основного состояния прибора ⁽²⁾.
- (*) **ПолноеСостояние()** : *Объект* - возвращает список состояний прибора (то есть мультисостояние).
- (*) **ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии прибора.
- (*) **Шлейфы()** : *Объект* - возвращает коллекцию шлейфов.
- (*) **Выходы()** : *Объект* - возвращает коллекцию выходов.
- (*) **Считыватели()** : *Объект* - возвращает коллекцию считывателей.
- (*) **Порт()** : *Объект* - возвращает объект *Порт*, к которому подключен прибор.
- (*) **Приборы()** : *Объект* - возвращает коллекцию подчиненных приборов (только для пультов C2000 и C2000M).
- (*) **СбросПрибора(ВидСброса)** - производит сброс прибора:
 - 1 - Сброс прибора по питанию
 - 2 - Сброс тревоги
 - 3 - Сброс счетчика непереданных событий (только для прибора C2000-ИТ).
- (*) **ВыдачаЗвуковогоСообщения(ТипСообщения)** - выдача звукового сообщения (сигнала) на приборе ⁽³⁾:
 - 0 - Выключение звука
 - 1 - Одиночный гудок
 - 2 - Двойной гудок
 - 3 - Тройной гудок
 - 4 - Длинный гудок
 - 5 - Прерывистый звуковой сигнал.

- (*) **ВыдатьТекстовоеСообщение**("ТекстСообщения") - выдать текстовое сообщение (только для клавиатур С2000-К).
- (*) **АктивироватьВоспроизведение**(*ЗадержкаВоспроизведения* , *ВремяВоспроизведения* , *НомерФрагмента*) - активация речевого воспроизведения ⁽⁴⁾.
- (*) **ДеактивироватьВоспроизведение**() - деактивация речевого воспроизведения ⁽⁴⁾.
- (*) **ВклРечевогоОповещения**() - запуск\выключение речевого оповещения для всех приборов типа «Рупор» и «Рупор исп. 01» для данного com-порта.
- (*) **ВыслатьТекст**("ФорматСтроки", "Текст") - Отобразить текст на приборе типа «Бегущая строка».
Формат строки (например, "\C0 R2 S20 P2 T10"), определяется протоколом Heliotron:
D0 - время,
D1 - дата,
D2 - день недели,
R0 - бегущая строка,
R1 - раскрывающаяся слева,
R2 - выплывающая из центра,
R3 - раскрывающаяся из центра,
R4 - бегущая вверх,
R5 - бегущая вниз,
R6 - замена,
C0 - красный цвет,
C1 - зеленый цвет,
C2 - оранжевый цвет,
C3 - переменный цвет,
P0 - без паузы,
P1 - пауза,
P2 - мигание,
P3 - смена цвета,
S0 ... S255 - время изменения картинки (8мс * N),
~FF 80 80 80 FF~ - bitmap,
T0 - T255 - время паузы в секундах.

Данная команда работает, только со строками компании Гелиотрон

- (*) **Телефоны**() : *Объект* - возвращает коллекцию телефонов, телефонов ассоциированных с устройством ⁽⁵⁾.
- (*) **ТранслироватьСобытие**(*КодСобытия* , *Элемент* , *Раздел* , *Сотрудник* , *ВремяИДата*) - трансляция в интерфейс события ⁽⁶⁾, где:
КодСобытия : *Число* - код события.
Элемент : *Объект* - элемент, на котором произошло событие. Может принимать значение, содержащее один из следующих типов объектов: «ПРИБОР», «ШЛЕЙФ», «ВЫХОД», «СЧИТЫВАТЕЛЬ».
Раздел : *Объект* - раздел или группа разделов, куда входит элемент. Может принимать значение, содержащее один из следующих типов объектов: «РАЗДЕЛ», «ГРУППАРАЗДЕЛОВ».
Сотрудник : *Объект* - сотрудник, который имеет отношение к событию (для событий управления и событий доступа).
ВремяИДата : *Число* - время и дата события.
- (*) **ТипЭлемента**() : *Строка* - возвращает тип, [см табл. 1](#).
- (*) **ТипЭлементаЧисло**() : *Число* - тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Типы приборов, поддерживаемые АРМ «Орион Про»: «С2000», «СИГНАЛ-20», «СИГНАЛ-20П», «С2000-СП1», «С2000-4», «С2000-К», «С2000-ИТ», «С2000-КДЛ», «С2000-БИ», «СИГНАЛ-20-02», «С2000-КС», «С2000-АСПТ», «С2000-КПБ», «С2000-2», «СИГНАЛ-20М», «РУПОР», «С2000-ПТ», «ПОТОК-3Н», «С2000-БИ ИСП. 01», «РУПОР ИСП. 01», «С2000-БКИ», «РИП-12 RS», «УО-4С», «С2000-ADEM», «УОП».

⁽²⁾ Возможные состояния приборов: «НЕТ КОНТАКТА» (*код 250*), «ТРЕВОГА ВЗЛОМА» (*код 149*), «КЗ ДПЛС» (*код 215*), «АВАРИЯ ДПЛС» (*код 222*), «НЕИСПРАВНОСТЬ ИСТОЧНИКА ПИТАНИЯ» (*код 198*), «АВАРИЯ СЕТИ 220» (*код 2*), «НЕИСПРАВНОСТЬ БАТАРЕИ» (*код 202*), «ВОССТАНОВЛЕНИЕ ВСКРЫТИЯ КОРПУСА» (*код 152*), «ВОССТАНОВЛЕНИЕ ДПЛС» (*код 47*), «ВОССТАНОВЛЕНИЕ ИСТОЧНИКА ПИТАНИЯ» (*код 199*), «ВОССТАНОВЛЕНИЕ СЕТИ 220» (*код 1*), «ВОССТАНОВЛЕНИЕ БАТАРЕИ» (*код 200*), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (*код 251*).

- (3) Приборы, поддерживающие данную функцию в настоящее время: «С2000М», «С2000-К», «С2000-КС», «С2000-2» и «С2000-4».
- (4) Приборы, поддерживающие данную функцию в настоящее время: «Рупор» и «Рупор исп. 01».
- (5) Приборы, поддерживающие данную функцию в настоящее время: «УОП».
- (6) Приборы, поддерживающие данную функцию в настоящее время: «УО-4С».

Шлейфы

- (*) **Количество()** : *Число* - возвращает количество элементов в коллекции.
- (*) **КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*) **Первый()** - переходит к первому шлейфу в коллекции.
- (*) **Следующий()** - переходит к следующему шлейфу в коллекции.
- (*) **Предыдущий()** - переходит к предыдущему шлейфу в коллекции.
- (*) **Последний()** - переходит к последнему шлейфу в коллекции.
- (*) **Текущий()** : *Объект* - возвращает текущий объект *Шлейф*.
- (*) **Элемент(Индекс)** : *Объект* - возвращает объект *Шлейф* по индексу объекта в коллекции.
- (*) **ПолучитьПоАдресу("АдресШлейфа")** : *Объект* - возвращает объект *Шлейф* по указанному адресу вида "X.Y.Z.S" (где X - номер порта, Y - адрес пульта, Z - адрес прибора, S - номер шлейфа).
- (*) **ПолучитьПоИндексу(ИндексШлейфа)** : *Объект* - возвращает объект *Шлейф* по индексу шлейфа.
- (*) **ПолучитьПоИД(ИДШлейфа)** : *Объект* - возвращает объект *Шлейф* по идентификатору шлейфа.
- (*) **ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).
- (*) **ТипЭлементаЧисло()** : *Число* - тип объекта в числовом значении, [см табл. 1](#).

Шлейф

- (*) **Название()** : *Строка* - возвращает название шлейфа.
- (*) **Адрес()** : *Строка* - возвращает адрес шлейфа.
- (*) **ИД()** : *Число* - возвращает ИД шлейфа.
- (*) **Индекс()** : *Число* - возвращает индекс шлейфа.
- (*) **Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*) **ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*) **ТипЭлемента()** : *Строка* - возвращает тип объекта (для данного объекта возвращается «ШЛЕЙФ»).
- (*) **Тип()** : *Строка* - возвращает тип шлейфа ⁽¹⁾.
- (*) **Состояние()** : *Строка* - возвращает состояние шлейфа ⁽²⁾.
- (*) **ЗначениеСостояния()** : *Число* - возвращает код основного состояния шлейфа ⁽²⁾.
- (*) **ПолноеСостояние()** : *Объект* - возвращает список состояний шлейфа (то есть мультисостояние).
- (*) **ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии шлейфа.
- (*) **Разделы()** : *Объект* - возвращает коллекцию разделов, в которые входит шлейф ⁽³⁾.
- (*) **Прибор()** : *Объект* - возвращает прибор, которому принадлежит шлейф.
- (*) **Взять()** - берет зону под охрану.
- (*) **Снять()** - снимает зону с охраны.
- (*) **УстановитьЗадержку(ЧислоСекунд)** - установить задержку взятия шлейфа ⁽⁴⁾.
- (*) **ЗадержкаВзятия()** : *Число* - возвращает установленную задержку взятия шлейфа.
- (*) **ОсталосьДоВзятия()** : *Число* - возвращает оставшееся время до взятия шлейфа.
- (*) **УстановитьДоВзятия(ЧислоСекунд)** - установить оставшееся время до взятия шлейфа (используется для продления задержки взятия шлейфа, в процессе задержки взятия шлейфа) ⁽⁴⁾.
- (*) **ВклАвтоматики()** - включает автоматику ⁽⁵⁾.
- (*) **ВыклАвтоматики()** - выключает автоматику ⁽⁵⁾.
- (*) **ПускАСПТ()** - запуск пожаротушения ⁽⁵⁾.
- (*) **СбросТревоги()** - сброс пуска пожаротушения ⁽⁵⁾.
- (*) **ЗапросАЦП()** : *Логическое* - посылка команды на запрос АЦП. Команда не возвращает значение АЦП!

(*)**ВернутьАЦП()**:Число-возвращает последний замер АЦП. Эта функция будет работать, только в том случае, если включен сбор статистики или перед ней осуществляется вызов ЗапросАЦП(), при этом результат возвращается асинхронно, через определенное количество Времени, поэтому для ожидания ответа уместно использовать метод задержка(от 3 секунд и выше), сценарий может выглядеть следующим образом:

ЗапросАЦП()

.....

Задержка(10);//ожидаю 10 секунд

ВернутьАЦП()

ТипЧисло():Число- возвращает тип шлейфа, [см таблица 2](#).

(1) Типы шлейфов, поддерживаемые АРМ «Орион Про»: «ОХРАННЫЙ», «ВХОДНОЙ», «ТРЕВОЖНАЯ КНОПКА», «ПОЖАРНЫЙ», «РУЧНОЙ ПУСК», «АДРЕСНО-АНАЛОГОВЫЙ ДЫМОВОЙ», «АДРЕСНО-АНАЛОГОВЫЙ ТЕПЛОВОЙ», «ВЛАЖНОСТНЫЙ», «ТЕХНОЛОГИЧЕСКИЙ», «РУЧНОЙ ПУСК (РУПОР)», «ЦЕПЬ ДС ДВЕРЕЙ», «РУЧНОЙ ПУСК (АСПТ)», «СДУ», «РЕЖИМ АВТОМАТИЧЕСКОГО ЗАПУСКА», «ИСТОЧНИК ОП», «ИСТОЧНИК РП», «РЕЖИМ ПРИБОРА», «ДИСТАНЦИОННЫЙ ПУСК», «КОНТРОЛЬ НЕИСПРАВНОСТИ АУП («МД»)), «СОСТОЯНИЕ УСТРОЙСТВА», «МАССА», «ДАВЛЕНИЕ», «ЗАПУСК ПЕРВОГО РАБОЧЕГО НАСОСА», «ПИТАНИЕ ПЕРВОГО РАБОЧЕГО НАСОСА», «АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ПЕРВОГО РАБОЧЕГО НАСОСА», «ЗАПУСК ВТОРОГО РАБОЧЕГО НАСОСА», «ПИТАНИЕ ВТОРОГО РАБОЧЕГО НАСОСА», «АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ВТОРОГО РАБОЧЕГО НАСОСА», «ЗАПУСК РЕЗЕРВНОГО НАСОСА», «ПИТАНИЕ РЕЗЕРВНОГО НАСОСА», «АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ РЕЗЕРВНОГО НАСОСА», «ЗАПУСК ЖОКЕЙ-НАСОСА», «ПИТАНИЕ ЖОКЕЙ-НАСОСА», «АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ЖОКЕЙ-НАСОСА», «ЗАКРЫТИЕ ЭЛЕКТРОЗАДВИЖКИ», «ОТКРЫТИЕ ЭЛЕКТРОЗАДВИЖКИ», «ПИТАНИЕ ЭЛЕКТРОЗАДВИЖКИ», «ОСНОВНОЙ ВВОД АВР», «РЕЗЕРВНЫЙ ВВОД АВР», «ДРЕНЧЕРНАЯ ЗАВЕСА», «ОСНОВНОЙ РЕЗЕРВУАР», «РЕЗЕРВНЫЙ РЕЗЕРВУАР», «ДРЕНАЖНЫЙ ПРИЯМОК», «РЕЖИМ ЗАПУСКА», «ДД ЗАПУСКА», «РУЧНОЙ ПУСК (ПОТОК)», «ДАВЛЕНИЕ В СИСТЕМЕ», «ИСТОЧНИК 26 В», «СОСТОЯНИЕ КЦ1», «СОСТОЯНИЕ КЦ2», «СОСТОЯНИЕ КЦ3», «СОСТОЯНИЕ КЦ4», «СОСТОЯНИЕ КЦ5», «СОСТОЯНИЕ КЦ6», «СОСТОЯНИЕ КЦ7», «СОСТОЯНИЕ КЦ8», «СОСТОЯНИЕ КЦ9», «СОСТОЯНИЕ КЦ10», «СОСТОЯНИЕ КЦ11», «СОСТОЯНИЕ КЦ12», «СОСТОЯНИЕ КЦ13», «СОСТОЯНИЕ КЦ14», «СОСТОЯНИЕ КЦ15», «СОСТОЯНИЕ КЦ16», «СОСТОЯНИЕ КЦ17», «СОСТОЯНИЕ КЦ18», «ВЫХОДНОЕ НАПРЯЖЕНИЕ», «ВЫХОДНОЙ ТОК», «ПРОВЕРКА АКБ», «ПРОВЕРКА ЗУ», «ПРОВЕРКА 220В», «ПРОГРАММИРУЕМЫЙ ТЕХНОЛОГИЧЕСКИЙ», «АДЕМСО (ПРИЕМНИК)», «АДЕМСО (РАДИОПОВТОРИТЕЛЬ)», «ПОЖАРНЫЙ АДРЕСНО-Пороговый», «ИСТОЧНИК ПИТАНИЯ 27В», «АГРЕГАТ 1», «АГРЕГАТ 2», «АГРЕГАТ 3», «АГРЕГАТ 4», «ДИСТАНЦИОННЫЙ ПУСК ПОТОК», «ВЫХОД Р1», «ВЫХОД Р2», «ВЫХОД Р3», «ВЫХОД Р4», «ЗОНЫ УОП».

(2) Возможные состояния шлейфов: «ТУШЕНИЕ» (код 144), «АВАРИЙНЫЙ ПУСК» (код 145), «ПУСК АСПТ» (код 146), «НЕУДАЧНЫЙ ПУСК ПТ» (код 139), «ЗАДЕРЖКА АВТОМАТИЧЕСКОГО ПУСКА» (код 141), «БЛОКИРОВКА ПУСКА» (код 147), «СБРОС ПУСКА АСПТ» (код 143), «ПОЖАР» (код 37), «ВНИМАНИЕ! ОПАСНОСТЬ ПОЖАРА» (код 44), «ТИХАЯ ТРЕВОГА» (код 58), «ТРЕВОГА ПРОНИКНОВЕНИЯ» (код 3), «ТРЕВОГА ВХОДНОЙ ЗОНЫ» (код 118), «ТРЕВОГА ВЗЛОМА» (код 149), «НЕТ КОНТАКТА» (код 250), «ШС ОТКЛЮЧЕН» (код 187), «ОТКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 192), «ПЕРЕГРУЗКА ИСТОЧНИКА ПИТАНИЯ» (код 194), «КОРОТКОЕ ЗАМЫКАНИЕ ДПЛС» (код 215), «АВАРИЯ ДПЛС» (код 222), «НЕКОРРЕКТНЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 224), «НЕУСТОЙЧИВЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 225), «ОШИБКА ПАРАМЕТРОВ ШС» (код 165), «НЕИСПРАВНОСТЬ ПОЖАРНОГО ОБОРУДОВАНИЯ» (код 41), «ОБРЫВ ШЛЕЙФА» (код 45), «КОРОТКОЕ ЗАМЫКАНИЕ» (код 214), «НЕУДАЧНОЕ ВЗЯТИЕ» (код 17), «НАРУШЕНИЕ СНЯТОЙ ЗОНЫ» (код 119), «ИДЕТ СНЯТИЕ» (код 244), «СНЯТ» (код 109), «ВОССТАНОВЛЕНИЕ СНЯТОЙ ЗОНЫ» (код 117), «НЕИСПРАВНОСТЬ БАТАРЕИ» (код 202), «НЕИСПРАВНОСТЬ ЗУ» (код 196), «АВАРИЯ СЕТИ» (код 2), «ЗАДЕРЖКА ВЗЯТИЯ» (код 23), «ИДЕТ ВЗЯТИЕ» (код 243), «ВЗЯТ» (код 24), «СБРОС ТРЕВОГИ» (код 110), «НЕИСПРАВНОСТЬ ИСТОЧНИКА ПИТАНИЯ» (код 198), «АВТОМАТИКА ВЫКЛЮЧЕНА» (код 142), «ВКЛЮЧЕНИЕ НАСОСА» (код 130), «ВЫКЛЮЧЕНИЕ НАСОСА» (код 131), «АВТОМАТИКА ВКЛЮЧЕНА» (код 148), «НАРУШЕНИЕ ТЕХНОЛОГИЧЕСКОГО ШС» (код 36), «ВОССТАНОВЛЕНИЕ ТЕХНОЛОГИЧЕСКОГО ШС» (код 35), «ПОЖАРНОЕ ОБОРУДОВАНИЕ В НОРМЕ» (код 39), «АВАРИЙНОЕ ПОВЫШЕНИЕ УРОВНЯ» (код 75), «АВАРИЙНОЕ ПОНИЖЕНИЕ УРОВНЯ» (код 77), «ПОВЫШЕНИЕ УРОВНЯ» (код 74), «ПОНИЖЕНИЕ УРОВНЯ» (код 71), «УРОВЕНЬ В НОРМЕ» (код 72), «НЕИСПРАВНОСТЬ ТЕРМОМЕТРА» (код 82), «ПОВЫШЕНИЕ ТЕМПЕРАТУРЫ» (код 76), «ПОНИЖЕНИЕ

ТЕМПЕРАТУРЫ» (код 206), «ТЕМПЕРАТУРА В НОРМЕ» (код 78), «ВОССТАНОВЛЕНИЕ ЗОНЫ КОНТРОЛЯ ВЗЛОМА» (код 152), «ВОССТАНОВЛЕНИЕ ВНУТРЕННЕЙ ЗОНЫ» (код 158), «ПОДКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 193), «ПЕРЕГРУЗКА ИСТОЧНИКА УСТРАНЕНА» (код 195), «ВОССТАНОВЛЕНИЕ ДПЛС» (код 47), «ВОССТАНОВЛЕНИЕ БАТАРЕИ» (код 200), «ВОССТАНОВЛЕНИЕ ЗУ» (код 197), «ВОССТАНОВЛЕНИЕ СЕТИ» (код 7), «ВОССТАНОВЛЕНИЕ ИСТОЧНИКА ПИТАНИЯ» (код 199), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (код 257), «ШС ПОДКЛЮЧЕН» (код 188), «НЕОПРЕДЕЛЕННОЕ» (код 0).

⁽³⁾ В настоящее время шлейф может входить только в один раздел. Поэтому в коллекции будет присутствовать только один объект.

⁽⁴⁾ Установленная задержка будет работать до закрытия Ядра (CoreOrion.exe), либо до перезагрузки Базы данных в Ядре.

⁽⁵⁾ Данные команды имеют смысл только для соответствующих шлейфов приборов С2000-АСПТ и Поток-3Н.

Выходы

(*) **Количество()** : *Число* - возвращает количество элементов в коллекции.

(*) **КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.

(*) **Первый()** - переходит к первому выходу.

(*) **Следующий()** - переходит к следующему выходу.

(*) **Предыдущий()** - переходит к предыдущему выходу.

(*) **Последний()** - переходит к последнему выходу.

(*) **Текущий()** : *Объект* - возвращает текущий объект *Выход*.

(*) **Элемент(Индекс)** : *Объект* - возвращает объект *Выход* по индексу объекта в коллекции.

(*) **ПолучитьПоАдресу("АдресВыхода")** : *Объект* - возвращает объект *Выход* по указанному адресу вида "X.Y.Z.S" (где X - номер порта, Y - адрес пульта, Z - адрес прибора, S - номер выхода).

(*) **ПолучитьПоИндексу(ИндексВыхода)** : *Объект* - возвращает объект *Выход* по индексу выхода.

(*) **ПолучитьПоИД(ИДШлейфа)** : *Объект* - возвращает объект *Выход* по идентификатору выхода.

Выход

(*) **Название()** : *Строка* - возвращает название выхода.

(*) **Адрес()** : *Строка* - возвращает адрес выхода.

(*) **ИД()** : *Число* - возвращает ИД выхода.

(*) **Индекс()** : *Число* - возвращает индекс выхода.

(*) **Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*) **ТипТега()** : *Строка* - возвращает тип значения тега объекта.

(*) **ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.

(*) **ТипЭлемента()** : *Строка* - возвращает тип объекта (для данного объекта возвращается «ВЫХОД»).

(*) **Тип()** : *Строка* - возвращает тип выхода ⁽¹⁾.

(*) **Состояние()** : *Строка* - возвращает состояние выхода ⁽²⁾.

(*) **ЗначениеСостояния()** : *Число* - возвращает код основного состояния выхода ⁽²⁾.

(*) **ПолноеСостояние()** : *Объект* - возвращает список состояний выхода (то есть мультисостояние).

(*) **ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии выхода.

(*) **Разделы()** : *Объект* - возвращает коллекцию разделов, в которые входит выход ⁽³⁾.

(*) **РазделыТактик()** : *Объект* - возвращает коллекцию связанных с выходом разделов.

(*) **Прибор()** : *Объект* - возвращает прибор, которому принадлежит выход.

(*) **ВозвратВИсходноеСостояние()** - переводит выход в исходное состояние.

(*) **Включить("ЗадержкаУправления")** - запускает для данного выхода тактику «Включить» с указанной *Задержкой управления* ^{(4) (5)}.

(*) **Выключить("ЗадержкаУправления")** - запускает для данного выхода тактику «Выключить» с указанной *Задержкой управления*.

(*) **ВключитьНаВремя("ЗадержкаУправления", "ВремяУправления")** - запускает для данного выхода тактику «Включить на время» с указанными

Задержкой управления и Временем управления.

- (*)**ВыключитьНаВремя**("ЗадержкаУправления", "ВремяУправления") - запускает для данного выхода тактику «Выключить на время» с указанными *Задержкой управления и Временем управления.*
- (*)**МигатьИзСостоянияВыключено**("ЗадержкаУправления", "0", "НомерМаскиМигания") - запускает для данного выхода тактику «Мигать из состояния выключено» с указанными *Задержкой управления и Номером Маски Мигания* ^{(4) (5)}.
- (*)**МигатьИзСостоянияВключено**("ЗадержкаУправления", "0", "НомерМаскиМигания") - запускает для данного выхода тактику «Мигать из состояния включено» с указанными *Задержкой управления и Номером Маски Мигания.*
- (*)**МигатьИзСостоянияВыключеноНаВремя**("ЗадержкаУправления", "ВремяУправления", "НомерМаскиМигания") - запускает для данного выхода тактику «Мигать из состояния выключено на время» с указанными *Задержкой управления, Временем управления и Номером Маски Мигания.*
- (*)**МигатьИзСостоянияВключеноНаВремя**("ЗадержкаУправления", "ВремяУправления", "НомерМаскиМигания") - запускает для данного выхода тактику «Мигать из состояния включено на время» с указанными *Задержкой управления, Временем управления и Номером Маски Мигания.*
- (*)**Лампа**() - запускает для данного выхода тактику «Системная Лампа» (только для прибора С2000-СП1 версии 1.20 и выше).
- (*)**ПЦН**() - запускает для данного выхода тактику «Системный ПЦН» (только для прибора С2000-СП1 версии 1.20 и выше).
- (*)**АСПТ**("0", "ЗадержкаУправления", "ВремяУправления") - запускает пожаротушение ⁽⁶⁾.
- (*)**ТипЭлемента**() : *Строка* - возвращает тип, [см табл. 1](#).

(*)**ТипЭлементаЧисло**() : Число-тип объекта в числовом значении, [см табл. 1](#).

(*)**ЗапросАЦП**() : **Логическое** -посылка команды на запрос АЦП. Команда не возвращает значение АЦП!

(*)**ВернутьАЦП**() : **Число**-возвращает последний замер АЦП. Эта функция будет работать, только в том случае, если включен сбор статистики или перед ней осуществляется вызов **ЗапросАЦП()**, при этом результат возвращается асинхронно, через определенное количество Времени, поэтому для ожидания ответа уместно использовать метод **задержка**(от 3 секунд и выше), сценарий может выглядеть следующим образом:

ЗапросАЦП()

.....

Задержка(10);//ожидаю 10 секунд

ВернутьАЦП()

ТипЧисло() : Число- возвращает тип шлейфа [см табл.3](#),

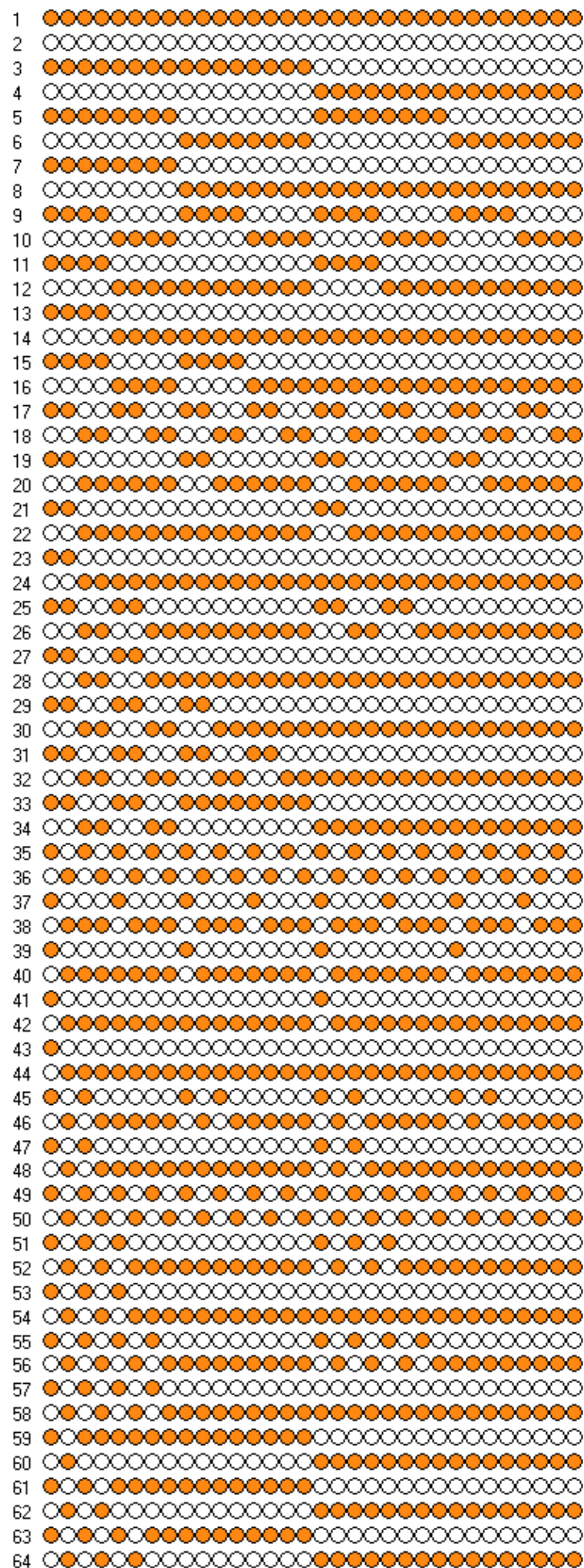
⁽¹⁾ Типы выходов, поддерживаемые АРМ «Орион Про»: «РЕЛЕ», «АДРЕСНЫЙ РЕЛЕЙНЫЙ МОДУЛЬ», «КОНТРОЛИРУЕМЫЙ ВЫХОД», «ВЫХОД КПБ (АСПТ)», «СО1 (УХОДИ)», «СО2 (НЕ ВХОДИ)», «СОЗ (АВТОМАТИКА ОТКЛЮЧЕНА)», «ЗО (СИРЕНА)», «ПУСКОВАЯ ЦЕПЬ», «ПУСК 1», «ПУСК 2», «ПУСК 3», «ПУСК 4», «РЕЧЕВОЕ ОПОВЕЩЕНИЕ».

⁽²⁾ Возможные состояния выходов: «НЕТ КОНТАКТА» (*Код 250*), «ОТКЛЮЧЕНИЕ ВЫХОДА» (*Код 126*), «ОБРЫВ ЦЕПИ» (*Код 121*), «КОРОТКОЕ ЗАМЫКАНИЕ ЦЕПИ» (*Код 122*), «ВОССТАНОВЛЕНИЕ ЦЕПИ» (*Код 123*), «ПОДКЛЮЧЕНИЕ ВЫХОДА» (*Код 127*), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (*Код 251*), «НЕОПРЕДЕЛЕННОЕ» (*Код 0*), «РЕЛЕ ВКЛЮЧЕНО» (*Код 401*), «РЕЛЕ ВЫКЛЮЧЕНО» (*Код 402*), «РЕЛЕ МИГАЕТ 3» (*Код 403*), «РЕЛЕ МИГАЕТ 4» (*Код 404*) ... «РЕЛЕ МИГАЕТ 64» (*Код 464*).

⁽³⁾ В настоящее время выход может входить только в один раздел. Поэтому в коллекции будет присутствовать только один объект.

⁽⁴⁾ Задержка и время управления, а также номер маски мигания задаются в кавычках, как текстовое значение.

⁽⁵⁾ Типы масок мигания, поддерживаемые АРМ «Орион Про» (1 кружок - 0,125 сек):



⁽⁶⁾ Имеет смысл использовать только для 32-го реле прибора «Поток-3Н» версии 1.02 и выше.

Считыватели

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому считывателю.
- (*)Следующий() - переходит к следующему считывателю.
- (*)Предыдущий() - переходит к предыдущему считывателю.
- (*)Последний() - переходит к последнему считывателю.
- (*)Текущий() : Объект - возвращает текущий объект Считыватель.
- (*)Элемент(Индекс) : Объект - возвращает объект Считыватель по индексу объекта в коллекции.
- (*)ПолучитьПоАдресу("АдресСчитывателя") : Объект - возвращает объект Считыватель по указанному адресу вида "X.Y.Z.S" (где X - номер порта, Y - адрес пульта, Z - адрес прибора, S - номер считывателя).
- (*)ПолучитьПоИндексу(ИндексСчитывателя) : Объект - возвращает объект Считыватель по индексу считывателя.
- (*)ПолучитьПоИД(ИДСчитывателя) : Объект - возвращает объект Считыватель по идентификатору считывателя.

Считыватель

- (*)Название() : Строка - возвращает название считывателя.
- (*)Адрес() : Строка - возвращает адрес считывателя.
- (*)ИД() : Число - возвращает ИД считывателя.
- (*)Индекс() : Число - возвращает индекс считывателя.
- (*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : Строка - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.
- (*)Состояние() : Строка - возвращает состояние считывателя ⁽¹⁾.
- (*)ЗначениеСостояния() : Число - возвращает код основного состояния считывателя ⁽¹⁾.
- (*)ПолноеСостояние() : Объект - возвращает список состояний считывателя (то есть мультисостояние).
- (*)ЕстьСостояние(КодСостояния) : Логическое - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии считывателя.
- (*)Прибор() : Объект - возвращает прибор, которому принадлежит считыватель.
- (*)ЗаблокироватьДоступ() - производит блокировку доступа со считывателя(ей) и «кнопки Выход» через дверь, управляемую данным считывателем.
- (*)ВосстановитьДоступ() - производит восстановление доступа со считывателя(ей) кнопки Выход через дверь, управляемую данным считывателем.
- (*)ОткрытиеДоступ() - производит открытие свободного доступа через дверь, управляемую данным считывателем.
- (*)ПредоставитьДоступ() - производит предоставление доступа через дверь, управляемую данным считывателем.
- (*)ЗаблокироватьКнопкуВыход() - производит блокировку доступа по кнопке «Выход» через дверь, управляемую данным считывателем.
- (*)РазблокироватьКнопкуВыход() - производит восстановление доступа по кнопке «Выход» через дверь, управляемую данным считывателем.
- (*)ТипЭлемента() : Строка - возвращает тип, [см тбл. 1](#).
- (*)ТипЭлементаЧисло() : Число-тип объекта в числовом значении, [см тбл. 1](#).
- (*)ЗаблокироватьСчитыватель() - производит блокировку доступа со считывателя . Блокировка кнопки ВЫХОД не производится!
- (*)РазблокироватьСчитыватель() - производит разблокировку доступа со считывателя. Разблокировка кнопки Выход не производится!

(1) Возможные состояния считывателей: «НЕТ КОНТАКТА» (Код 250), «ДОСТУП ЗАКРЫТ» (Код 25), «ДОСТУП ОТКРЫТ» (Код 219), «ДОСТУП ВОССТАНОВЛЕН» (Код 30), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (Код 251), «НЕОПРЕДЕЛЕННОЕ» (Код 0).

Видеосистемы

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первой видеосистеме в коллекции.
- (*)Следующий() - переходит к следующей видеосистеме в коллекции.
- (*)Предыдущий() - переходит к предыдущей видеосистеме в коллекции.
- (*)Последний() - переходит к последней видеосистеме в коллекции.
- (*)Текущий() : Объект - возвращает текущий объект *Видеосистема*.
- (*)Элемент(Индекс) : Объект - возвращает объект *Видеосистема* по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(ИндексВидеосистемы) : Объект - возвращает объект *Видеосистема* по индексу видеосистемы.
- (*)ПолучитьПоИД(ИДВидеосистемы) : Объект - возвращает объект *Видеосистема* по идентификатору видеосистемы.

Видеосистема

- (*)Название() : Строка - возвращает название видеосистемы.
- (*)ИД() : Число - возвращает ИД видеосистемы.
- (*)Индекс() : Число - возвращает индекс видеосистемы.
- (*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : Строка - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.
- (*)Компьютер() : Объект - возвращает компьютер, которому принадлежит видеосистема.
- (*)Камеры() : Объект - возвращает коллекцию камер видеосистемы.
- (*)ТипЭлемента() : Строка - возвращает тип, [см табл. 1](#).
- (*)ТипЭлементаЧисло() : Число-тип объекта в числовом значении, [см табл. 1](#).

Камеры

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первой камере в коллекции.
- (*)Следующий() - переходит к следующей камере в коллекции.
- (*)Предыдущий() - переходит к предыдущей камере в коллекции.
- (*)Последний() - переходит к последней камере в коллекции.
- (*)Текущий() : Объект - возвращает текущий объект *Камера*.
- (*)Элемент(Индекс) : Объект - возвращает объект *Камера* по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(ИндексКамеры) : Объект - возвращает объект *Камера* по индексу камеры.
- (*)ПолучитьПоИД(ИДКамеры) : Объект - возвращает объект *Камера* по идентификатору камеры.

Камера

- (*)Название() : Строка - возвращает название камеры.
- (*)ИД() : Число - возвращает ИД камеры.
- (*)Индекс() : Число - возвращает индекс камеры.
- (*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : Строка - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.
- (*)ТипЭлемента() : Строка - возвращает тип объекта (для данного объекта возвращается «КАМЕРА»).

- (*)**Состояние()** : *Строка* - возвращает основное состояние камеры ⁽¹⁾.
- (*)**ЗначениеСостояния()** : *Число* - возвращает код основного состояния камеры ⁽¹⁾.
- (*)**ПолноеСостояние()** : *Объект* - возвращает список состояний шлейфа (то есть мультисостояние).
- (*)**ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии камеры.
- (*)**Разделы()** : *Объект* - возвращает коллекцию разделов, в которые входит камера ⁽²⁾.
- (*)**Видеосистема()** : *Объект* - возвращает видеосистему, которой принадлежит камера.
- (*)**Взять()** - берет камеру под охрану.
- (*)**Снять()** - снимает камеру с охраны.
- (*)**ВключитьДетектор()** - включает детектор движения камеры.
- (*)**ВыключитьДетектор()** - выключает детектор движения камеры.
- (*)**НачатьЗапись()** - включает запись с камеры.
- (*)**ЗакончитьЗапись()** - выключает запись с камеры.
- (*)**Показать()** - показывает монитор с изображением с камеры.
- (*)**Скрыть()** - скрывает монитор с изображением с камеры.
- (*)**Повернуть(НомерПресет)** - управление трансфокатором.
- (*)**ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).
- (*)**ТипЭлементаЧисло()**: *Число*-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Возможные состояния камер: «ТРЕВОГА ДЕТЕКТОРА ДВИЖЕНИЯ» (*Код 3*), «КАМЕРА ОТКЛЮЧЕНА» (*Код 187*), «КАМЕРА СНЯТА С ОХРАНЫ» (*Код 109*), «КАМЕРА НА ОХРАНЕ» (*Код 24*), «СРАБАТЫВАНИЕ ДЕТЕКТОРА ДВИЖЕНИЯ» (*Код 1224*), «ДЕТЕКТОР ДВИЖЕНИЯ ВЫКЛЮЧЕН» (*Код 1216*), «ДЕТЕКТОР ДВИЖЕНИЯ ВЫКЛЮЧЕН» (*Код 1215*), «ЗАПИСЬ ЗАКОНЧЕНА» (*Код 1207*), «ЗАПИСЬ НАЧАТА» (*Код 1206*), «КАМЕРА ВКЛЮЧЕНА» (*Код 188*), «НЕОПРЕДЕЛЕННОЕ» (*Код 0*).

⁽²⁾ В настоящее время камера может входить только в один раздел. Поэтому в коллекции будет присутствовать только один объект.

ГруппыРазделов

- (*)**Количество()** : *Число* - возвращает количество элементов в коллекции.
- (*)**КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)**Первый()** - переходит к первой группе разделов.
- (*)**Следующий()** - переходит к следующей группе разделов.
- (*)**Предыдущий()** - переходит к предыдущей группе разделов.
- (*)**Последний()** - переходит к последней группе разделов.
- (*)**Текущий()** : *Объект* - возвращает текущий объект *ГруппаРазделов*.
- (*)**Элемент(Индекс)** : *Объект* - возвращает объект *ГруппаРазделов* по индексу объекта в коллекции.
- (*)**ПолучитьПоИндексу(ИндексГруппыРазделов)** : *Объект* - возвращает объект *ГруппаРазделов* по индексу группы разделов.
- (*)**ПолучитьПоИД(ИДГруппыРазделов)** : *Объект* - возвращает объект *ГруппаРазделов* по идентификатору группы разделов.

ГруппаРазделов

- (*)**Название()** : *Строка* - возвращает название группы разделов.
- (*)**ИД()** : *Число* - возвращает ИД группы разделов.
- (*)**Индекс()** : *Число* - возвращает индекс группы разделов.
- (*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*)**Состояние()** : *Строка* - возвращает состояние группы разделов ⁽¹⁾.
- (*)**ЗначениеСостояния()** : *Число* - возвращает код основного состояния группы разделов ⁽¹⁾.
- (*)**ПолноеСостояние()** : *Объект* - возвращает список состояний группы разделов (то есть мультисостояние).
- (*)**ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии группы

разделов.

(*)**Разделы()** : *Объект* - возвращает коллекцию разделов, входящих в данную группу разделов.

(*)**Взять()** - производит взятие группы разделов на охрану.

(*)**Снять()** - производит снятие группы разделов с охраны.

(*)**ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).

(*)**ТипЭлементаЧисло()**:*Число*-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Возможные состояния группы разделов состоят из состояния всех зон, входящих в разделы данной группы разделов: приборов, шлейфов, реле и камер (все состояния данных объектов приведены выше). Основные состояния группы разделов: «ТУШЕНИЕ» (код 144), «АВАРИЙНЫЙ ПУСК» (код 145), «ПУСК АСПТ» (код 146), «НЕУДАЧНЫЙ ПУСК ПТ» (код 139), «ЗАДЕРЖКА АВТОМАТИЧЕСКОГО ПУСКА» (код 141), «БЛОКИРОВКА ПУСКА» (код 147), «СБРОС ПУСКА АСПТ» (код 143), «ДВА ПОЖАРА» (код 255), «ПОЖАР» (код 37), «ВНИМАНИЕ! ОПАСНОСТЬ ПОЖАРА» (код 44), «ТИХАЯ ТРЕВОГА» (код 58), «ТРЕВОГА ПРОНИКНОВЕНИЯ» (код 3), «ТРЕВОГА ВХОДНОЙ ЗОНЫ» (код 118), «ТРЕВОГА ВЗЛОМА» (код 149), «НЕТ КОНТАКТА» (код 250), «ОТКЛЮЧЕН» (код 187), «ОТКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 192), «ПЕРЕГРУЗКА ИСТОЧНИКА ПИТАНИЯ» (код 194), «КОРОТКОЕ ЗАМЫКАНИЕ ДПЛС» (код 215), «АВАРИЯ ДПЛС» (код 222), «НЕКОРРЕКТНЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 224), «НЕУСТОЙЧИВЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 225), «ОШИБКА ПАРАМЕТРОВ» (код 165), «НЕИСПРАВНОСТЬ ПОЖАРНОГО ОБОРУДОВАНИЯ» (код 47), «ОБРЫВ ШЛЕЙФА» (код 45), «КОРОТКОЕ ЗАМЫКАНИЕ» (код 214), «НЕУДАЧНОЕ ВЗЯТИЕ» (код 17), «НАРУШЕНИЕ СНЯТОЙ ЗОНЫ» (код 119), «ИДЕТ СНЯТИЕ» (код 244), «СНЯТ» (код 109), «ВОССТАНОВЛЕНИЕ СНЯТОЙ ЗОНЫ» (код 117), «НЕИСПРАВНОСТЬ БАТАРЕИ» (код 202), «НЕИСПРАВНОСТЬ ЗУ» (код 196), «АВАРИЯ СЕТИ» (код 2), «ЗАДЕРЖКА ВЗЯТИЯ» (код 23), «ИДЕТ ВЗЯТИЕ» (код 243), «ВЗЯТ» (код 24), «СРАБАТЫВАНИЕ ДЕТЕКТОРА ДВИЖЕНИЯ» (код 1224), «ДЕТЕКТОР ДВИЖЕНИЯ ВЫКЛЮЧЕН» (код 1216), «ДЕТЕКТОР ДВИЖЕНИЯ ВКЛЮЧЕН» (код 1215), «СБРОС ТРЕВОГИ» (код 110), «НЕИСПРАВНОСТЬ ИСТОЧНИКА ПИТАНИЯ» (код 198), «АВТОМАТИКА ВЫКЛЮЧЕНА» (код 142), «ВКЛЮЧЕНИЕ НАСОСА» (код 130), «ВЫКЛЮЧЕНИЕ НАСОСА» (код 131), «АВТОМАТИКА ВКЛЮЧЕНА» (код 148), «НАРУШЕНИЕ ТЕХНОЛОГИЧЕСКОГО» (код 36), «ВОССТАНОВЛЕНИЕ ТЕХНОЛОГИЧЕСКОГО» (код 35), «ОТКЛЮЧЕНИЕ ВЫХОДА» (код 126), «ОБРЫВ ЦЕПИ НАГРУЗКИ ВЫХОДА» (код 127), «КОРОТКОЕ ЗАМЫКАНИЕ НАГРУЗКИ ВЫХОДА» (код 122), «ПОЖАРНОЕ ОБОРУДОВАНИЕ В НОРМЕ» (код 39), «АВАРИЙНОЕ ПОВЫШЕНИЕ УРОВНЯ» (код 75), «АВАРИЙНОЕ ПОНИЖЕНИЕ УРОВНЯ» (код 77), «ПОВЫШЕНИЕ УРОВНЯ» (код 74), «ПОНИЖЕНИЕ УРОВНЯ» (код 71), «УРОВЕНЬ В НОРМЕ» (код 72), «НЕИСПРАВНОСТЬ ТЕРМОМЕТРА» (код 82), «ПОВЫШЕНИЕ ТЕМПЕРАТУРЫ» (код 76), «ПОНИЖЕНИЕ ТЕМПЕРАТУРЫ» (код 206), «ТЕМПЕРАТУРА В НОРМЕ» (код 78), «ЗАПИСЬ ЗАКОНЧЕНА» (код 1207), «ЗАПИСЬ НАЧАТА» (код 1206), «ВОССТАНОВЛЕНИЕ ЦЕПИ НАГРУЗКИ» (код 123), «ВОССТАНОВЛЕНИЕ ЗОНЫ КОНТРОЛЯ ВЗЛОМА» (код 152), «ВОССТАНОВЛЕНИЕ ВНУТРЕННЕЙ ЗОНЫ» (код 158), «ПОДКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 193), «ПЕРЕГРУЗКА ИСТОЧНИКА УСТРАНЕНА» (код 195), «ВОССТАНОВЛЕНИЕ ДПЛС» (код 47), «ВОССТАНОВЛЕНИЕ БАТАРЕИ» (код 200), «ВОССТАНОВЛЕНИЕ ЗУ» (код 197), «ВОССТАНОВЛЕНИЕ СЕТИ» (код 1), «ВОССТАНОВЛЕНИЕ ИСТОЧНИКА ПИТАНИЯ» (код 199), «ПОДКЛЮЧЕН» (код 188), «ПОДКЛЮЧЕНИЕ ВЫХОДА» (код 127), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (код 251), «НЕОПРЕДЕЛЕННОЕ» (код 0).

Разделы

(*)**Количество()** : *Число* - возвращает количество элементов в коллекции.

(*)**КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.

(*)**Первый()** - переходит к первому разделу.

(*)**Следующий()** - переходит к следующему разделу.

(*)**Предыдущий()** - переходит к предыдущему разделу.

(*)**Последний()** - переходит к последнему разделу.

(*)**Текущий()** : *Объект* - возвращает текущий объект *Раздел*.

(*)**Элемент(Индекс)** : *Объект* - возвращает объект *Раздел* по индексу объекта в коллекции.

(*)**ПолучитьПоИндексу(ИндексРаздела)** : *Объект* - возвращает объект *Раздел* по индексу раздела.

(*)**ПолучитьПоИД(ИДРаздела)** : *Объект* - возвращает объект *Раздел* по идентификатору раздела.

Раздел

- (*)**Название()** : *Строка* - возвращает название раздела.
 - (*)**Индекс()** : *Число* - возвращает индекс раздела.
 - (*)**ИД()** : *Число* - возвращает ИД раздела.
 - (*)**Тег()** : *Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
 - (*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.
 - (*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
 - (*)**Состояние()** : *Строка* - возвращает основное состояние раздела ⁽¹⁾.
 - (*)**ЗначениеСостояния()** : *Число* - возвращает код основного состояния раздела ⁽¹⁾.
 - (*)**ПолноеСостояние()** : *Объект* - возвращает список состояний раздела (то есть мультисостояние).
 - (*)**ЕстьСостояние(КодСостояния)** : *Логическое* - возвращает результат проверки присутствия состояния с указанным кодом в полном состоянии раздела.
 - (*)**Зоны()** : *Объект* - возвращает коллекцию зон, входящих в раздел: шлейфов, выходов, камер и приборов.
 - (*)**Выходы()** : *Объект* - возвращает коллекцию выходов, входящих в раздел.
 - (*)**ГруппыРазделов()** : *Объект* - возвращает коллекцию групп разделов, в которые входит раздел.
 - (*)**Взять()** - производит взятие раздела на охрану.
 - (*)**Снять()** - производит снятие раздела с охраны.
 - (*)**ВклАвтоматики()** - включает автоматику ⁽²⁾.
 - (*)**ВклАвтоматики()** - выключает автоматику ⁽²⁾.
 - (*)**ПускАСПТ()** - запуск пожаротушения ⁽²⁾.
 - (*)**СбросТревоги()** - сброс пуска пожаротушения ⁽²⁾.
 - (*)**ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).
- СбросАСПТ(): Логическое- сброс запуска АСПТ, в случае нахождения в разделе зон удаленного сброса, от приюров С2000-АСПТ\Поток-3Н.
- (*)**ТипЭлементаЧисло()**: Число-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Возможные состояния раздела состоят из состояния всех зон, входящих раздел: приборов, шлейфов, реле и камер (все состояния данных объектов приведены выше). Основные состояния раздела: «ТУШЕНИЕ» (код 144), «АВАРИЙНЫЙ ПУСК» (код 145), «ПУСК АСПТ» (код 146), «НЕУДАЧНЫЙ ПУСК ПТ» (код 139), «ЗАДЕРЖКА АВТОМАТИЧЕСКОГО ПУСКА» (код 141), «БЛОКИРОВКА ПУСКА» (код 147), «СБРОС ПУСКА АСПТ» (код 143), «ДВА ПОЖАРА» (код 255), «ПОЖАР» (код 37), «ВНИМАНИЕ! ОПАСНОСТЬ ПОЖАРА» (код 44), «ТИХАЯ ТРЕВОГА» (код 58), «ТРЕВОГА ПРОНИКНОВЕНИЯ» (код 3), «ТРЕВОГА ВХОДНОЙ ЗОНЫ» (код 118), «ТРЕВОГА ВЗЛОМА» (код 149), «НЕТ КОНТАКТА» (код 250), «ОТКЛЮЧЕН» (код 187), «ОТКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 192), «ПЕРЕГРУЗКА ИСТОЧНИКА ПИТАНИЯ» (код 194), «КОРОТКОЕ ЗАМЫКАНИЕ ДПЛС» (код 215), «АВАРИЯ ДПЛС» (код 222), «НЕКОРРЕКТНЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 224), «НЕУСТОЙЧИВЫЙ ОТВЕТ УСТРОЙСТВА В ДПЛС» (код 225), «ОШИБКА ПАРАМЕТРОВ» (код 165), «НЕИСПРАВНОСТЬ ПОЖАРНОГО ОБОРУДОВАНИЯ» (код 41), «ОБРЫВ ШЛЕЙФА» (код 45), «КОРОТКОЕ ЗАМЫКАНИЕ» (код 214), «НЕУДАЧНОЕ ВЗЯТИЕ» (код 17), «НАРУШЕНИЕ СНЯТОЙ ЗОНЫ» (код 119), «ИДЕТ СНЯТИЕ» (код 244), «СНЯТ» (код 109), «ВОССТАНОВЛЕНИЕ СНЯТОЙ ЗОНЫ» (код 117), «НЕИСПРАВНОСТЬ БАТАРЕИ» (код 202), «НЕИСПРАВНОСТЬ ЗУ» (код 196), «АВАРИЯ СЕТИ» (код 2), «ЗАДЕРЖКА ВЗЯТИЯ» (код 23), «ИДЕТ ВЗЯТИЕ» (код 243), «ВЗЯТ» (код 24), «СРАБАТЫВАНИЕ ДЕТЕКТОРА ДВИЖЕНИЯ» (код 1224), «ДЕТЕКТОР ДВИЖЕНИЯ ВЫКЛЮЧЕН» (код 1216), «ДЕТЕКТОР ДВИЖЕНИЯ ВКЛЮЧЕН» (код 1215), «СБРОС ТРЕВОГИ» (код 110), «НЕИСПРАВНОСТЬ ИСТОЧНИКА ПИТАНИЯ» (код 198), «АВТОМАТИКА ВЫКЛЮЧЕНА» (код 142), «ВКЛЮЧЕНИЕ НАСОСА» (код 130), «ВЫКЛЮЧЕНИЕ НАСОСА» (код 131), «АВТОМАТИКА ВКЛЮЧЕНА» (код 148), «НАРУШЕНИЕ ТЕХНОЛОГИЧЕСКОГО» (код 36), «ВОССТАНОВЛЕНИЕ ТЕХНОЛОГИЧЕСКОГО» (код 35), «ОТКЛЮЧЕНИЕ ВЫХОДА» (код 126), «ОБРЫВ ЦЕПИ НАГРУЗКИ ВЫХОДА» (код 121), «КОРОТКОЕ ЗАМЫКАНИЕ НАГРУЗКИ ВЫХОДА» (код 122), «ПОЖАРНОЕ ОБОРУДОВАНИЕ В НОРМЕ» (код 39), «АВАРИЙНОЕ ПОВЫШЕНИЕ УРОВНЯ» (код 75), «АВАРИЙНОЕ ПОНИЖЕНИЕ УРОВНЯ» (код 77), «ПОВЫШЕНИЕ УРОВНЯ» (код 74), «ПОНИЖЕНИЕ УРОВНЯ» (код 71), «УРОВЕНЬ В НОРМЕ» (код 72), «НЕИСПРАВНОСТЬ ТЕРМОМЕТРА» (код 82), «ПОВЫШЕНИЕ ТЕМПЕРАТУРЫ» (код 76), «ПОНИЖЕНИЕ ТЕМПЕРАТУРЫ» (код 206), «ТЕМПЕРАТУРА В НОРМЕ» (код 78), «ЗАПИСЬ ЗАКОНЧЕНА» (код 1207), «ЗАПИСЬ НАЧАТА» (код 1206), «ВОССТАНОВЛЕНИЕ ЦЕПИ НАГРУЗКИ» (код 123), «ВОССТАНОВЛЕНИЕ ЗОНЫ КОНТРОЛЯ ВЗЛОМА» (код 152), «ВОССТАНОВЛЕНИЕ

ВНУТРЕННЕЙ ЗОНЫ» (код 158), «ПОДКЛЮЧЕНИЕ ВЫХОДНОГО НАПРЯЖЕНИЯ ИСТОЧНИКА ПИТАНИЯ» (код 193), «ПЕРЕГРУЗКА ИСТОЧНИКА УСТРАНЕНА» (код 195), «ВОССТАНОВЛЕНИЕ ДПЛС» (код 47), «ВОССТАНОВЛЕНИЕ БАТАРЕИ» (код 200), «ВОССТАНОВЛЕНИЕ ЗУ» (код 197), «ВОССТАНОВЛЕНИЕ СЕТИ» (код 1), «ВОССТАНОВЛЕНИЕ ИСТОЧНИКА ПИТАНИЯ» (код 199), «ПОДКЛЮЧЕН» (код 188), «ПОДКЛЮЧЕНИЕ ВЫХОДА» (код 127), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (код 251), «НЕОПРЕДЕЛЕННОЕ» (код 0).

⁽²⁾ Данные команды имеют смысл только для разделов с соответствующими шлейфами приборов С2000-АСПТ и Поток-ЗН.

Двери

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первой двери.
- (*)Следующий() - переходит к следующей двери.
- (*)Предыдущий() - переходит к предыдущей двери.
- (*)Последний() - переходит к последней двери.
- (*)Текущий() : Объект - возвращает текущий объект Дверь.
- (*)Элемент(Индекс) : Объект - возвращает объект Дверь по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(ИндексДвери) : Объект - возвращает объект Дверь по индексу.
- (*)ПолучитьПоИД(ИДДвери) : Объект - возвращает объект Дверь по ИД.

Дверь

- (*)Название() : Строка - возвращает название двери.
- (*)Индекс() : Число - возвращает индекс двери.
- (*)ИД() : Число - возвращает ИД двери.
- (*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : Строка - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.
- (*)Состояние() : Строка - возвращает основное состояние двери ⁽¹⁾.
- (*)ЗначениеСостояния() : Число - возвращает код основного состояния двери ⁽¹⁾.
- (*)Режим() : Строка - возвращает режим работы двери ⁽²⁾.
- (*)СчитывательНаВход() : Объект - возвращает объект «Считыватель» с которого осуществляется доступ на вход.
- (*)СчитывательНаВыход() : Объект - возвращает объект «Считыватель» с которого осуществляется доступ на выход.
- (*)ЗонаНаВход() : Объект - возвращает объект «Зона доступа» на вход.
- (*)ЗонаНаВыход() : Объект - возвращает объект «Зона доступа» на выход.
- (*)ТипЭлемента() : Строка - возвращает тип, [см табл. 1](#).
- (*)ТипЭлементаЧисло() : Число-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Возможные состояния двери: «ДОСТУП ЗАКРЫТ» (код 25), «ДОСТУП ОТКЛОНЕН» (код 26), «ДВЕРЬ ВЗЛОМАНА» (код 27), «ДОСТУП ПРЕДОСТАВЛЕН» (код 28), «ЗАПРЕТ ДОСТУПА» (код 29), «ВОССТАНОВЛЕНИЕ ДОСТУПА» (код 30), «ВОССТАНОВЛЕНИЕ ЦЕЛОСТНОСТИ ДВЕРИ» (код 31), «ПРОХОД» (код 32), «ДВЕРЬ ЗАБЛОКИРОВАНА» (код 33), «СВОБОДНЫЙ ДОСТУП ОТКРЫТ» (код 219), «ВОССТАНОВЛЕНИЕ КОНТАКТА» (код 251), «НЕТ КОНТАКТА» (код 250), «НЕОПРЕДЕЛЕННОЕ» (код -1).

⁽²⁾ Возможные режимы двери: «ПРОХОД», «ВХОД», «ВЫХОД», «ВХОД/ВЫХОД».

ЗоныДоступа

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первой зоне доступа.
- (*)Следующий() - переходит к следующей зоне доступа.
- (*)Предыдущий() - переходит к предыдущей зоне доступа.
- (*)Последний() - переходит к последней зоне доступа.
- (*)Текущий() : Объект - возвращает текущий объект ЗонаДоступа.

- (*)**Элемент(Индекс)** : *Объект* - возвращает объект *ЗонаДоступа* по индексу объекта в коллекции.
- (*)**ПолучитьПоИндексу(ИндексЗоныДоступа)** : *Объект* - возвращает объект *ЗонаДоступа* по индексу.
- (*)**ПолучитьПоИД(ИДЗоныДоступа)** : *Объект* - возвращает объект *ЗонаДоступа* по ИД.

ЗонаДоступа

- (*)**Название()** : *Строка* - возвращает название зоны доступа.
- (*)**Индекс()** : *Число* - возвращает индекс зоны доступа.
- (*)**ИД()** : *Число* - возвращает ИД зоны доступа.
- (*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*)**Сотрудники()** : *Объект* - возвращает коллекцию сотрудников, находящихся в зоне доступа.
- (*)**КоличествоСотрудниковВЗонеДоступа()** : *Число* - возвращает количество сотрудников находящихся в зоне доступа.
- (*)**КоличествоСотрудниковПоУровню(ИДГруппыДоступа)** : *Число* - возвращает количество сотрудников с указанной группой доступа находящихся в зоне доступа.
- (*)**КоличествоСотрудниковПоПодразделению(ИДПодразделения)** : *Число* - возвращает количество сотрудников указанного подразделения находящихся в зоне доступа.
- (*)**УровеньДоступаПоследнего()** : *Число* - возвращает ID уровня доступа последнего вошедшего в зону доступа сотрудника.
- (*)**ПарольПоследнего()** : *Число* - возвращает ID пароля последнего вошедшего в зону доступа сотрудника.
- (*)**ПодразделениеПоследнегоСотрудника()** : *Число* - возвращает ID подразделения последнего вошедшего в зону доступа сотрудника.
- (*)**СотрудникПоследнего()** : *Число* - возвращает ID последнего вошедшего в зону доступа сотрудника.
- (*)**ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).
- (*)**ТипЭлементаЧисло()** : *Число* - тип объекта в числовом значении, [см табл. 1](#).

ГруппыДоступа

- (*)**Количество()** : *Число* - возвращает количество элементов в коллекции.
- (*)**КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)**Первый()** - переходит к первой группе доступа.
- (*)**Следующий()** - переходит к следующей группе доступа.
- (*)**Предыдущий()** - переходит к предыдущей группе доступа.
- (*)**Последний()** - переходит к последней группе доступа.
- (*)**Текущий()** : *Объект* - возвращает текущий объект *ГруппаДоступа*.
- (*)**Элемент(Индекс)** : *Объект* - возвращает объект *ГруппаДоступа* по индексу объекта в коллекции.
- (*)**ПолучитьПоИндексу(ИндексГруппыДоступа)** : *Объект* - возвращает объект *ГруппаДоступа* по индексу. ⁽¹⁾
- (*)**ПолучитьПоИД(ИДГруппыДоступа)** : *Объект* - возвращает объект *ГруппаДоступа* по ИД.

⁽¹⁾ Индекс для объекта «ГруппаДоступа» равен порядковому номеру группы доступа при загрузке групп доступа в оперативную память компьютера.

ГруппаДоступа

- (*)**Название()** : *Строка* - возвращает название группы доступа.
- (*)**Индекс()** : *Число* - возвращает индекс группы доступа ⁽¹⁾.
- (*)**ИД()** : *Число* - возвращает ИД группы доступа.
- (*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра

опроса.)

- (*) **ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*) **Двери()** : *Объект* - возвращает коллекцию дверей, полномочия на управление которыми присутствуют в группе доступа ⁽²⁾.
- (*) **Разделы()** : *Объект* - возвращает коллекцию разделов, полномочия на управление которыми присутствуют в группе доступа. ⁽²⁾.
- (*) **ГруппыРазделов()** : *Объект* - возвращает коллекцию групп разделов, полномочия на управление которыми присутствуют в группе доступа. ⁽²⁾.

⁽¹⁾ Индекс для объекта «ГруппаДоступа» равен порядковому номеру группы доступа при загрузке групп доступа в оперативную память компьютера.

⁽²⁾ Для коллекций дверей, разделов и групп разделов, получаемых из уровня доступа, доступны дополнительно следующие методы:

- (*) **ВременноеОкно(Индекс)** : *Объект* - возвращает объект временное окно, назначенное двери, разделу или группе разделов с тем же индексом в коллекции.
- (*) **АктивностьВременногоОкна(Индекс)** : *Логическое* - возвращает активность временного окна, назначенного двери, разделу или группе разделов с тем же индексом в коллекции.
- (*) **Полномочия(Индекс)** : *Число* - возвращает числовое значение полномочий, назначенное двери, разделу или группе разделов с тем же индексом в коллекции.
Для дверей:
 - 0 - Проход,
 - 1 - Вход,
 - 2 - Выход,
 - 3 - Вход\Выход.Для разделов и групп разделов:
 - 0 - Просмотр,
 - 1 - Взятие,
 - 2 - Снятие,
 - 3 - Взятие\Снятие.

(*) **ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).

(*) **ТипЭлементаЧисло()** : *Число* - тип объекта в числовом значении, [см табл. 1](#).

Подразделения

- (*) **Количество()** : *Число* - возвращает количество элементов в коллекции.
- (*) **КоличествоЭлементов()** : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*) **Первый()** - переходит к первому подразделению.
- (*) **Следующий()** - переходит к следующему подразделению.
- (*) **Предыдущий()** - переходит к предыдущему подразделению.
- (*) **Последний()** - переходит к последнему подразделению.
- (*) **Текущий()** : *Объект* - возвращает текущий объект подразделению.
- (*) **Элемент(Индекс)** : *Объект* - возвращает объект *Подразделение* по индексу объекта в коллекции.
- (*) **ПолучитьПоИД(ИДТелефона)** : *Объект* - возвращает объект *Подразделение* по ИД подразделения.

Подразделение

- (*) **Название()** : *Строка* - возвращает название подразделения.
- (*) **ИД()** : *Число* - возвращает ИД подразделения.
- (*) **Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*) **ТипТега()** : *Строка* - возвращает тип значения тега объекта.
- (*) **ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.
- (*) **Сотрудники()** : *Объект* - возвращает коллекцию сотрудников данного подразделения.
- (*) **ГрафикРаботы()** : *Объект* - возвращает объект «ГруппаДоступа» для данного подразделения.

(*)ТипЭлемента() : Строка - возвращает тип, [см табл. 1](#).

(*)ТипЭлементаЧисло():Число-тип объекта в числовом значении, [см табл. 1](#).

Сотрудники

(*)Количество() : Число - возвращает количество элементов в коллекции.

(*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.

(*)Первый() - переходит к первому сотруднику.

(*)Следующий() - переходит к следующему сотруднику.

(*)Предыдущий() - переходит к предыдущему сотруднику.

(*)Последний() - переходит к последнему сотруднику.

(*)Текущий() : Объект - возвращает текущий объект Сотрудник.

(*)Элемент(Индекс) : Объект - возвращает объект Сотрудник по индексу объекта в коллекции.

(*)ПолучитьПоИндексу(ИндексСотрудника) : Объект - возвращает объект Сотрудник по индексу. ⁽¹⁾

(*)ПолучитьПоИД(ИДСотрудника) : Объект - возвращает объект Сотрудник по ИД.

(*)ПолучитьПоТабНомеру("ТабНомерСотрудника") : Объект - возвращает объект Сотрудник по табельному номеру сотрудника ⁽²⁾.

⁽¹⁾ Индекс для объекта «Сотрудник» равен порядковому номеру сотрудника при загрузке сотрудников в оперативную память компьютера.

⁽²⁾ Табельный номер должен быть уникальным. В случае, если у нескольких сотрудников одинаковый табельный номер, то возвращается первый найденный сотрудник.

Сотрудник

(*)Индекс() : Число - возвращает индекс сотрудника ⁽¹⁾.

(*)ИД() : Число - возвращает ИД сотрудника.

(*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*)ТипТега() : Строка - возвращает тип значения тега объекта.

(*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.

(*)ТабНомер() : Строка - возвращает табельный номер сотрудника.

(*)Имя() : Строка - возвращает имя сотрудника.

(*)Фамилия() : Строка - возвращает фамилию сотрудника.

(*)Отчество() : Строка - возвращает отчество сотрудника.

(*)ФИО() : Строка - возвращает ФИО сотрудника.

(*)ЗонаДоступа() : Объект - возвращает объект Зона доступа в которой находится сотрудник.

(*)Пароли() : Объект - возвращает коллекцию паролей, принадлежащих данному сотруднику.

(*)ДатаРождения() : Число - возвращает дату рождения сотрудника.

(*)Подразделение() : Объект - возвращает подразделение в которое входит сотрудник.

(*)ТипЭлемента() : Строка - возвращает тип, [см табл. 1](#).

(*)ТипЭлементаЧисло():Число-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Индекс для объекта «Сотрудник» равен порядковому номеру сотрудника при загрузке сотрудников в оперативную память компьютера.

Пароли ⁽¹⁾

(*)Количество() : Число - возвращает количество элементов в коллекции.

(*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.

(*)Первый() - переходит к первому паролю.

(*)Следующий() - переходит к следующему паролю.

(*)Предыдущий() - переходит к предыдущему паролю.

(*)Последний() - переходит к последнему паролю.

- (*)Текущий() : Объект - возвращает текущий объект Пароль.
- (*)Элемент(Индекс) : Объект - возвращает объект Пароль по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(ИндексПароля) : Объект - возвращает объект Пароль по индексу. ⁽²⁾
- (*)ПолучитьПоИД(ИДПароля) : Объект - возвращает объект Пароль по индексу.
- (*)ПолучитьПоКоду("КодПароля") : Объект - возвращает объект Пароль по коду ключа\карточки или пин-коду.

⁽¹⁾ В коллекцию паролей входят все ключи\карты и «Пин-коды». «Пароли для программ» в коллекцию паролей не входят.

⁽²⁾ Индекс для объекта «Пароль» равен порядковому номеру пароля при загрузке паролей в оперативную память компьютера.

Пароль

- (*)Индекс() : Число - возвращает индекс пароля. ⁽¹⁾
- (*)ИД() : Число - возвращает ИД пароля.
- (*)Тег() : *Любой тип - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : Строка - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(Значение) - помещает значение любого типа в тег объекта.
- (*)СтроковыйПароль() : Строка - возвращает строковое представление пароля.
- (*)Сотрудник() : Объект - возвращает объект Сотрудник, которому принадлежит пароль.
- (*)ГруппаДоступа() : Объект - возвращает объект ГруппаДоступа, назначенная данному паролю.
- (*)СрокНачалаДействия() : Строка - возвращает срок начала действия пароля.
- (*)СрокОкончанияДействия() : Строка - возвращает срок окончания действия пароля.
- (*)Заблокировать() - заблокировать ключ.
Идентификаторы блокируются как централизованные, так и локальные. Изменения в Базе данных не происходят, но в Ядре опроса и приборах идентификатор блокируется, т.е. идентификаторы блокируются до момента синхронизации. После блокировки формируется специальное событие о централизованной блокировке.
- (*)Разблокировать() - разблокировать ключ.
- (*)ПроверитьБлокировку() : Логическое - возвращает состояние идентификатора: заблокирован или разблокирован в приборах.
- (*)УстановленаБлокировка() : Логическое - возвращает значение флага «Ключ заблокирован» идентификатора в Базе данных: установлен или нет.
- (*)УстановленВСтопЛист() : Логическое - возвращает значение флага «Стоп-лист» идентификатора в Базе данных: установлен или нет.
- (*)ТипЭлемента() : Строка - возвращает тип, [см табл. 1](#).
- (*)ТипЭлементаЧисло() : Число-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Индекс для объекта «Пароль» равен порядковому номеру пароля при загрузке паролей в оперативную память компьютера.

Скрипты

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : Число - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому скрипту.
- (*)Следующий() - переходит к следующему скрипту.
- (*)Предыдущий() - переходит к предыдущему скрипту.
- (*)Последний() - переходит к последнему скрипту.
- (*)Текущий() : Объект - возвращает текущий объект Скрипт.
- (*)Элемент(Индекс) : Объект - возвращает объект Скрипт по индексу объекта в коллекции.
- (*)ПолучитьПоИндексу(ИндексСкрипта) : Объект - возвращает объект Скрипт по индексу скрипта. ⁽¹⁾
- (*)ПолучитьПоИД(ИДСкрипта) : Объект - возвращает объект Скрипт по ИД

скрипта.

⁽¹⁾ Индекс для объекта «Скрипт» равен порядковому номеру скрипта при загрузке скриптов в оперативную память компьютера.

Скрипт

(*)**Название()** : *Строка* - возвращает название скрипта.

(*)**ИД()** : *Число* - возвращает ИД скрипта.

(*)**Индекс()** : *Число* - возвращает индекс скрипта ⁽¹⁾.

(*)**Тег()** : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)

(*)**ТипТега()** : *Строка* - возвращает тип значения тега объекта.

(*)**ПоместитьВТег(Значение)** - помещает значение любого типа в тег объекта.

(*)**ВыполнитьСценарий(Задержка)** - выполняется скрипт с указанной в секундах задержкой.

(*)**ЗадержкаСценария()** : *Число* - возвращает установленную для скрипта задержку. ⁽²⁾

(*)**УстановитьЗадержку(Задержка)** - установить задержку для скрипта. ⁽²⁾

(*)**Заблокировать(ИДСкрипта)** : *Логическое* - заблокировать сценарий, если была отправлена команда, то запуск сценария с указанным ID не произойдет. Если сценарий выполнялся он не будет остановлен!

(*)**Остановить(ИДСкрипта)** : *Логическое* - останавливает сценарий, если он выполняется. Сценарий может быть остановлен, если внутри его используются:

1)команды:

Задержка, Пауза.

2)методы объектов:

ЗапросОператору;

ВводЗначения;

ВЫБОРОПЕРАТОРА.

В противном случае выполняемый сценарий остановить не возможно.

(*)**Разблокировать(ИДСкрипта)** : *Логическое* - разблокировать сценарий, снимает блокировку с заблокированного на управление сценария.

(*)**Заблокирован** : *Логическое*, флаг сигнализирующий состояние сценария, да-заблокирован, Нет- разблокирован.

(*)**Выполняется** : *Логическое*, флаг сигнализирующий состояние запуска сценария, да-запущен-выполняется, Нет- не запущен.

(*)**ТипЭлемента()** : *Строка* - возвращает тип, [см табл. 1](#).

(*)**ТипЭлементаЧисло()**:*Число*-тип объекта в числовом значении, [см табл. 1](#).

⁽¹⁾ Индекс для объекта «Скрипт» равен порядковому номеру скрипта при загрузке скриптов в оперативную память компьютера.

⁽²⁾ Данные функции используются вместе с функцией «ВыполнитьСценарий» до истечения задержки запуска скрипта..

ПолноеСостояние

(*)**Количество ()** : *Число* - возвращает количество состояний в списке состояний.

(*)**Элемент(Номер)** : *Число* - возвращает текстовое описание состояния по номеру состояния в списке состояний.

(*)**ЗначениеСостояния(Номер)** : *Число* - возвращает код состояния по номеру состояния в списке состояний.

(*)**ПолучитьПоСостоянию(КодСостояния)** : *Число* - возвращает номер состояния в списке состояний по коду состояния.

(*)**ТипЭлемента(Номер)** : *Строка* - возвращает тип элемента, которому принадлежит состояние с указанным номером. ⁽¹⁾

⁽¹⁾ Данная функция доступна только для объекта «ПолноеСостояние» полученного из объекта «Раздел» (возвращаемые типы: 3 - шлейф, 4 - прибор, 5 - камера, 9 - выход) или «ГруппаРазделов» (возвращаемый тип: 2 - раздел).

Телефоны

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому телефону.
- (*)Следующий() - переходит к следующему телефону.
- (*)Предыдущий() - переходит к предыдущему телефону.
- (*)Последний() - переходит к последнему телефону.
- (*)Текущий() : *Объект* - возвращает текущий объект *Телефон*.
- (*)Элемент(*Индекс*) : *Объект* - возвращает объект *Телефон* по индексу объекта в коллекции.
- (*)ПолучитьПоИД(*ИДТелефона*) : *Объект* - возвращает объект *Телефон* по ИД телефона.
- (*)ПолучитьПоНомеруТелефона ("*НомерТелефона*") : *Объект* - возвращает объект *Телефон* по номеру телефона.

Телефон

- (*)Телефон() : *Строка* - возвращает номер телефона.
- (*)ИД() : *Число* - возвращает ИД телефона.
- (*)Тег() : **Любой тип* - возвращает значение тега объекта. (Тег - какая-либо пользовательская информация любого типа, сохраняемая до закрытия Ядра опроса.)
- (*)ТипТега() : *Строка* - возвращает тип значения тега объекта.
- (*)ПоместитьВТег(*Значение*) - помещает значение любого типа в тег объекта.
- (*)ТипЭлемента() : *Строка* - возвращает тип, [см табл. 1](#).
- (*)ТипЭлементаЧисло():Число-тип объекта в числовом значении, [см табл. 1](#).

СобытиеСкрипта ⁽¹⁾

- (*)Событие() : *Объект* - возвращает объект «Событие».
- (*)ВремяДата() : *Число* - возвращает время и дату события.
- (*)Прибор() : *Объект* - возвращает прибор, на котором произошло событие.
- (*)Элемент() : *Объект* - возвращает объект («Шлейф», «Реле», «Считыватель», «Камера»), которому принадлежит событие.
- (*)Объект() : *Объект* - возвращает раздел или группу разделов, зоне которого принадлежит событие.
- (*)Пароль() : *Объект* - возвращает пароль.
- (*)Дверь() : *Объект* - возвращает дверь для событий СКД.
- (*)РежимПрохода() : *Число* - возвращает режим прохода для событий СКД (0 - проход, 1 - вход, 2 - выход).
- (*)ЗонаДоступа() : *Объект* - возвращает зону доступа для событий СКД.

⁽¹⁾ Объект «СобытиеСкрипта» возвращается только при вызове системной функции «ПолучитьСобытие».

Событие

- (*)ИД() : *Число* - возвращает код события.
- (*)Название() : *Строка* - возвращает название события.

ПочтовыеЯщики

- (*)Количество() : Число - возвращает количество элементов в коллекции.
- (*)КоличествоЭлементов() : *Число* - возвращает количество «ненулевых» элементов коллекции.
- (*)Первый() - переходит к первому объекту(ПочтовыйЯщик).
- (*)Следующий() - переходит к следующему объекту(ПочтовыйЯщик).
- (*)Предыдущий() - переходит к предыдущему объекту(ПочтовыйЯщик).
- (*)Последний() - переходит к последнему объекту(ПочтовыйЯщик).
- (*)Текущий() : *Объект* - возвращает текущий объект объекту(ПочтовыйЯщик).
- (*)Элемент(*Индекс*) : *Объект* - возвращает объект *Сотрудник* по индексу объекта в коллекции.

(*)**ПолучитьПочтовыйЯщик** (*ИмяПочтовогоЯщика: Строка*) : Объект(ПочтовыйЯщик)- возвращает объект Почтовый ящик по названию почтового сервера
ДобавитьПочту(*ИмяПочтовогоЯщика:Строка; Логин:Строка; Пароль:Строка; Порт:Число;УдалениеПриВычитывании:Логическое;SSL:Логическое*):Логическое, подключение к почтовому ящику. Необходимо для удаленного управления взятием-снятием разделов, с помощью email. **Внимание! Рекомендуем заводить для этого способа управления, отдельный email, и не использовать личный или корпоративный почтовый сервер, который используется в работе.**
ИмяПочтовогоЯщика - имя почтового сервера, например «smtp.mail.ru»
Логин- логин для соединения с сервером, должен быть заранее зарегистрирован
Пароль - пароль логина для соединения с сервером, логин и пароль должны быть заранее зарегистрированы.
Порт- порт по которому происходит подключение к почтовому серверу, по умолчанию 110.
УдалениеПриВычитывании-удалять письмо при вычитывании с сервера. Если письма не будут удалять с сервера, то время на подключение к почтовому серверу может быть значительным. Рекомендуется выставлять, как Да. Письма вычитанные с сервера, исчезнут из почтового ящика навсегда.
SSL- флаг включения SSL шифрования, по умолчанию включен. Зависит от требований предъявляемым к соединению владельцами почтовых серверов.
Почта(): Объект(КоллекцияПочта), возвращает коллекцию подключений к почтовым серверам.
УдалитьПочту(*НаименованиеПочтовогоЯщика:Строка*)- удалить почтовый ящик из списка

ПочтовыйЯщик

(*)**ИД()** : *Число* - возвращает ИД почтового.
 (*)**ПолучитьПочту()** :Логическое - запускает чтение почты с сервера
 (*)**ПаузаПриемаПочты (Пауза:Число)** :Логическое - остановка приема почты, на указанное значение секунд(переменная Пауза), в случае передачи 0, устанавливает чтение почты. По умолчанию **Пауза=0**.
Остановлен():Логическое -проверка, остановлено ли чтение писем из почтового ящика; Да-остановлен;Нет-не остановлен.
РазрешитьПочту(*НаименованиеПочтовогоАдреса:Строка*)-разрешение на прием почты с заданного почтового адреса. Это разрешение позволяет управлять взятием-снятием с помощью посылки электронных писем.
ЗапретитьПочту(*НаименованиеПочтовогоАдреса:Строка*)-запретить прием почты с заданного адреса.
ПроверитьНаРазрешение(*НаименованиеПочтовогоАдреса:Строка*):Логическое - проверка на разрешение управления с почтового адреса

Для управления взятием-снятием необходимо, осуществить следующие действия:

- 1.Подключиться к почтовому серверу-**ПочтовыеЯщики,ДобавитьПочту(..)**;
- 2.Дать разрешение на управление с тех почтовых адресов, с которых разрешено удаленное управление ПочтовыйЯщик. **РазрешитьПочту('InfoControl@mail.ru')**;
- 3.Активировать получение почты **ПочтовыеЯщики.ПолучитьПочту**
- 4.С адреса InfoControl@mail.ru послать письмо в тело которого включить управляющие команды. EXECUTE_SCRIPT [id скрипта, который вы можете увидеть на вкладке «Сценарии управления» в АБД]

[Пример как регистрировать почтовые адреса, для последующего управления запуском сценариев.](#)

Пример регистрации почтового сервера для последующего управления.

Ниже приведен пример сценария, подсоединения к почтовому ящику, и разрешение на управление с почтовых адресов: MyName@bolid.ru, mmm@bolid.ru,

Сценарий

Переменные

Перем ПочтовыйЯщик5;

Перем Почта5;

Перем РабочееМесто5;

Перем РабочиеМеста5;

Перем Компьютер5;

Перем Компьютеры5;

Перем ПочтовыйЯщик4;
Перем Почта4;
Перем РабочееМесто4;
Перем РабочиеМеста4;
Перем Компьютер4;
Перем Компьютеры4;
Перем ПочтовыйЯщик3;
Перем Почта3;
Перем РабочееМесто3;
Перем РабочиеМеста3;
Перем Компьютер3;
Перем Компьютеры3;
Перем ПочтовыйЯщик2;
Перем Почта2;
Перем РабочееМесто2;
Перем РабочиеМеста2;
Перем Компьютер2;
Перем Компьютеры2;
Перем Логическое1;
Перем Почта1;
Перем РабочееМесто1;
Перем РабочиеМеста1;
Перем Компьютер1;
Перем Компьютеры1;
НачалоСценария

Попытка

Компьютеры1 = СоздатьОбъект("Компьютеры");
Компьютер1 = Компьютеры1.ПолучитьПоИд(1);
РабочиеМеста1 = Компьютер1.РабочиеМеста();
РабочееМесто1 = РабочиеМеста1.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто1) Тогда
Почта1 = РабочееМесто1.Почта();
Логическое1 = Логическое(0);
Почта1.ДобавитьПочту("mail-serv.company.ru", "admin", "orion", 110, Логическое1, 0);
КонецЕсли;
Исключение Сообщить("Произошло исключение", 0); КонецПопытки;

Попытка

Компьютеры2 = СоздатьОбъект("Компьютеры");
Компьютер2 = Компьютеры2.ПолучитьПоИд(1);
РабочиеМеста2 = Компьютер2.РабочиеМеста();
РабочееМесто2 = РабочиеМеста2.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто2) Тогда
Почта2 = РабочееМесто2.Почта();
ПочтовыйЯщик2 = Почта2.ПолучитьПочту("mail-serv.company.ru ");
Если не ПустоеЗначение(ПочтовыйЯщик2) Тогда
ПочтовыйЯщик2.РазрешитьПочту("mmm@bolid.ru");
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение", 0); КонецПопытки;

Попытка

Компьютеры3 = СоздатьОбъект("Компьютеры");
Компьютер3 = Компьютеры3.ПолучитьПоИд(1);
РабочиеМеста3 = Компьютер3.РабочиеМеста();
РабочееМесто3 = РабочиеМеста3.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто3) Тогда
Почта3 = РабочееМесто3.Почта();
ПочтовыйЯщик3 = Почта3.ПолучитьПочту("mail-serv.company.ru ");
Если не ПустоеЗначение(ПочтовыйЯщик3) Тогда
ПочтовыйЯщик3.РазрешитьПочту("MyName@bolid.ru");

КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;

Попытка
Компьютеры5 = СоздатьОбъект("Компьютеры");
Компьютер5 = Компьютеры5.ПолучитьПоИд(1);
РабочиеМеста5 = Компьютер5.РабочиеМеста();
РабочееМесто5 = РабочиеМеста5.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто5) Тогда
Почта5 = РабочееМесто5.Почта();
ПочтовыйЯщик5 = Почта5.ПолучитьПочту("mail-serv.company.ru ");
Если не ПустоеЗначение(ПочтовыйЯщик5) Тогда
ПочтовыйЯщик5.ПринятьПочту();
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;
КонецСценария

Ниже приведен пример электронного письма запуска сценария ID=123

From: [MyName@bolid.ru]
Sent: Thursday, July 23, 2015 12:20 PM
To: orion@company.ru
Subject: ORION_MAIL_COMMAND

Execute_script 123

Таблица 1. Список объектов

Id	Наименование	Текстовое обозначение
2	Раздел	РАЗДЕЛ
3	Зоны	ШЛЕЙФ
4	Устройства	ПРИБОР
5	Видеокамеры	КАМЕРА
6	Пароль	ПАРОЛЬ
7	Группы доступа	ГРУППАДОСТУПА
8	Считыватели	СЧИТЫВАТЕЛЬ
9	Реле и управляемые выходы	ВЫХОД
11	Зона доступа	ЗОНАДОСТУПА
12	Точка доступа	ТОЧКАДОСТУПА
14	Компьютер	КОМПЬЮТЕР
16	Порты на компьютере	ПОРТ
17	Группа разделов	ГРУППАРАЗДЕЛОВ
20	Персонал	ПЕРСОНАЛ
22	Подразделение	ПОДРАЗДЕЛЕНИЕ
23	Телефонный номер	ТЕЛЕФОН
24	Канал распознавания	КАНАЛРАСПОЗНАВАНИЯ
25	Видеосистемы	
48	События	
70	Сценарий	
71	Рабочие места	
72	Рабочее место	
73	Компьютеры	
74	Компьютер	
150	Ключница	
151	Пеналы	
257	Биометрический контроллер	
512	Устройство бегущая строка от	

513	Автомобили	

Таблица 2. Список типов шлейфов

Номер	Наименование типа
1	Охранный
2	Пожарный
3	Тревожная кнопка
4	Технологический
5	Входной
6	Адресно-аналоговый дымовой
7	Адресно-аналоговый тепловой
9	Цепь ДС дверей
10	Ручной пуск (АСПТ)
11	Дистанционный пуск
12	Режим автоматического запуска
13	Масса
14	Давление
15	СДУ
16	Источник ОП
17	Источник РП
19	Режим прибора
20	Запуск первого рабочего насоса
21	Питание первого рабочего насоса
22	Автоматическое управление первого рабочего насоса
23	Запуск второго рабочего насоса
24	Питание второго рабочего насоса
25	Автоматическое управление второго рабочего насоса
26	Запуск резервного насоса
27	Питание резервного насоса
28	Автоматическое управление резервного насоса
29	Запуск жockey-насоса
30	Питание жockey-насоса
31	Автоматическое управление жockey-насоса
32	Закрытие электрозадвижки
33	Открытие электрозадвижки
34	Питание электрозадвижки
35	Основной ввод АВР
36	Резервный ввод АВР
37	Дренчерная завеса
38	Основной резервуар
39	Резервный резервуар
40	Дренажный приямок
41	Режим запуска
42	ДД запуска
43	Ручной пуск (Поток)
45	Давление в системе
48	Источник 26 В
49	Состояние КЦ1
50	Состояние КЦ2
51	Состояние КЦ3
52	Состояние КЦ4
53	Состояние КЦ5
54	Состояние КЦ6
55	Состояние КЦ7
56	Состояние КЦ8
57	Состояние КЦ9
58	Состояние КЦ10
59	Состояние КЦ11
60	Состояние КЦ12
61	Состояние КЦ13

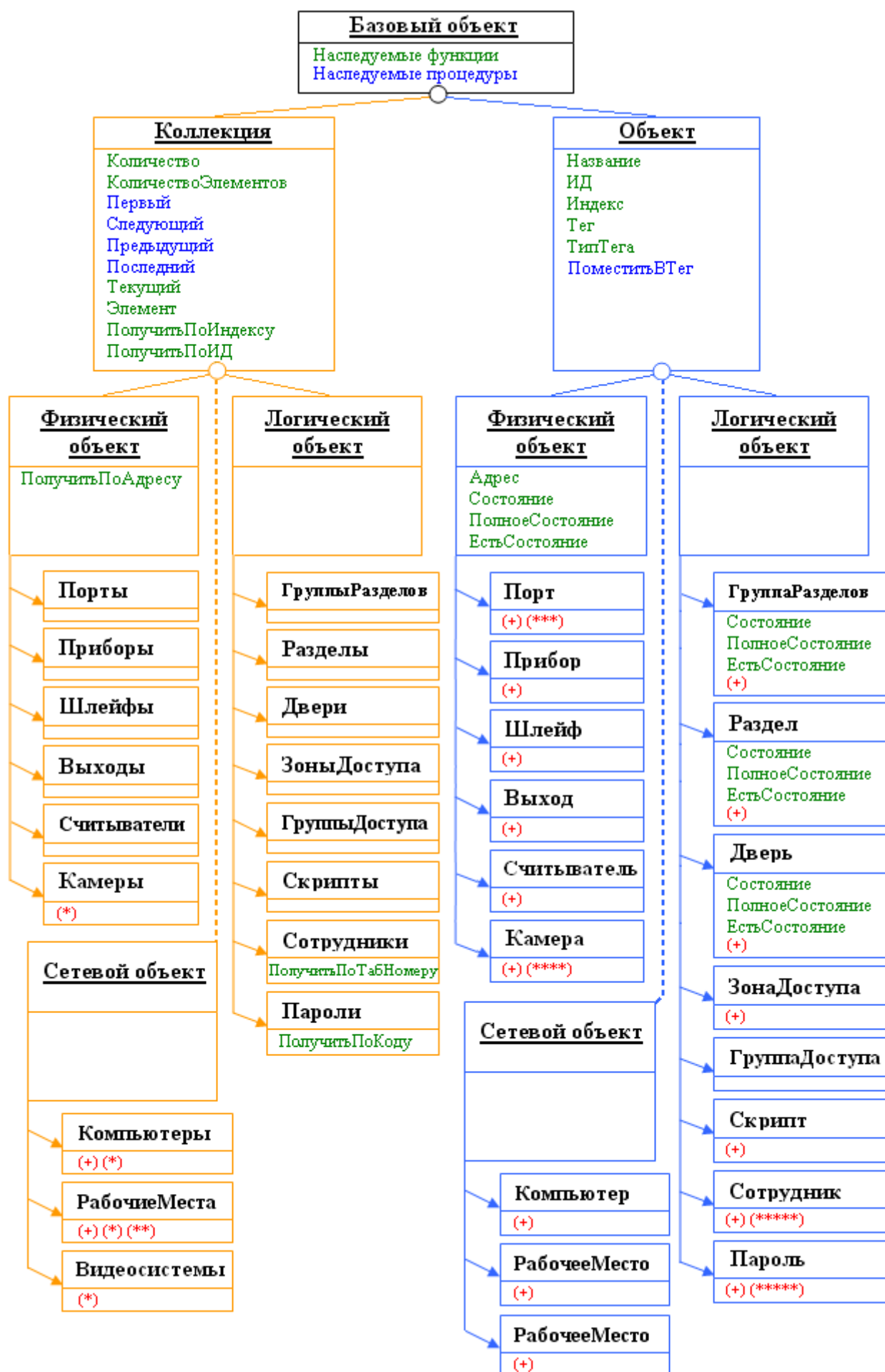
62	Состояние КЦ14
63	Состояние КЦ15
64	Состояние КЦ16
65	Состояние КЦ17
66	Состояние КЦ18
67	Ручной пуск (Рупор)
68	Ручной пуск
69	Состояние устройства
70	Контроль неисправности АУП («МД»)
71	Выходное напряжение
72	Выходной ток
73	Проверка АКБ
74	Проверка ЗУ
75	Проверка 220В
76	Программируемый технологический
77	Ademco (Приемник)
78	Ademco (Радиоповторитель)
79	Пожарный адресно-пороговый
80	Влажностный
81	Источник питания 27В
82	Агрегат 1
83	Агрегат 2
84	Агрегат 3
85	Агрегат 4
86	Дистанционный пуск Поток
87	Выходы Р1
88	Выходы Р2
89	Выходы Р3
90	Выходы Р4
91	Зоны УОП
92	Контроль ШС
93	Контроль источника ОП(220В)
94	Контроль ЗУ
95	Контроль источника РП(АКБ)
96	Контроль дистанционного запуска РО
97	Контроль состояния прибора
98	Концевой выключатель
99	Анчар-40
100	тантал-200
101	Тантал-600
102	Мурена
103	Водосигнальный
104	Зоны удаленных объектов

Таблица 3. Список типов выходов

Номер	Наименование типа
0	Реле
1	Контролируемый выход"
2	СО1 (УХОДИ)
3	СО2 (НЕ ВХОДИ)
4	СО3 (Автоматика отключена)
5	ЗО (Сирена)
6	Пусковая цепь
7	Пуск 1
8	Пуск 2
9	Пуск 3
10	Пуск 4
11	Адресный релейный модуль
12	Выход КПБ (АСПТ)
13	Речевое оповещение
14	Контролируемое реле

15	Клапан
16	Адресный выход управления клапаном

3.3. [Дерево объектов](#)

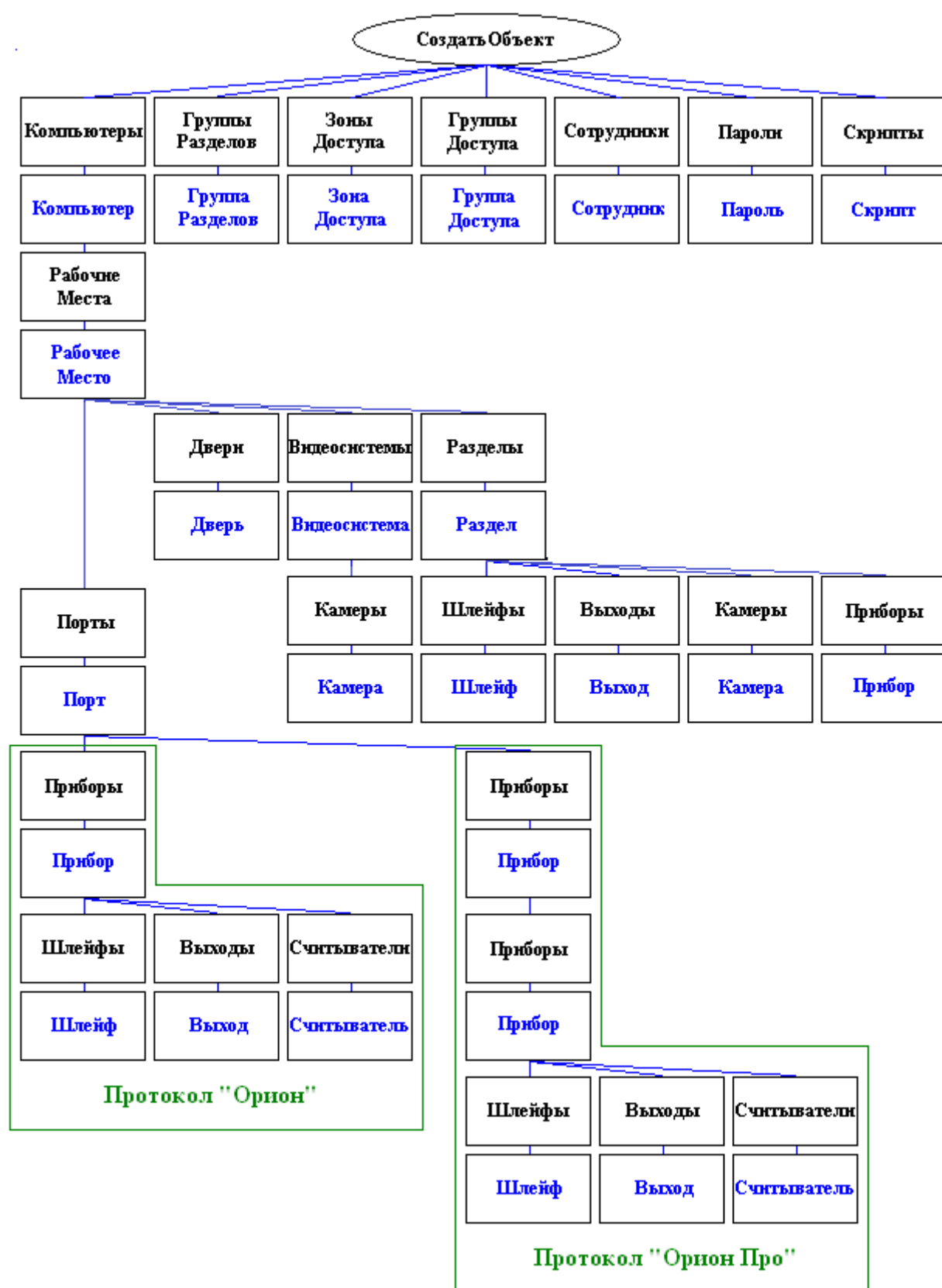


- (+)**Объект имеет дополнительно свой набор функций и процедур.**
- (*)**Блокирована функция «ПолучитьПоАдресу».**
- (**)**Блокированы функции «ПолучитьПоИД» и «ПолучитьПоИндексу».**
- (***)**Блокированы функции «Состояние», «ПолноеСостояние» и «ЕстьСостояние».**
- (****)**Блокирована функция «Адрес».**
- (*****)**Блокирована функция «Название».**

Внимание!

В случае если в системе используется одно Ядро опроса, либо в системе с несколькими Ядрами опроса сценарий запускается по событию, то можно получать коллекции всех типов объектов (кроме «Рабочие места») при помощи функции «СоздатьОбъект».

Если в системе с несколькими Ядрами опроса сценарий запускается по горячей клавише, из дерева управления, или по расписанию, то коллекции объектов типа «РабочиеМеста», «Порты», «Приборы», «Шлейфы», «Выходы», «Считыватели», «Видеосистемы», «Камеры», «Разделы» и «Двери» рекомендуется получать, двигаясь по объектам начиная от объекта «Компьютер»:



4. Шаблонные сценарии.

4.1. Узел «Прибор»

Номер	Наименование	Параметры	Описание
1	Выдать текстовое сообщение	1.Компьютер на котором находится устройство 2.Прибор - Прибор на который высылаем сообщение	Транслировать текстовое сообщение на прибор С2000-К

		3.Сообщение, не более 16 символов	
2	Активировать Воспроизведение	1.Компьютер на котором находится устройство 2.Прибор - Прибор на которым будем управлять 3.Задержка управления, в секундах. Задержка воспроизведения звукового фрагмента 4.Время воспроизведения, звукового фрагмента, в секундах 5.Номер звукового фрагмента	Активируется воспроизведение звуковых сообщений, записанных в Рупор. Активация не означает воспроизведение фрагмента, прибор переводится в состояние готовности воспроизведения. Для воспроизведения фрагментов необходимо осуществить запуск воспроизведения, с помощью команды «Пуск\Стоп речевого оповещения.
3	Деактивировать Воспроизведение	1.Компьютер на котором находится устройство 2.Прибор - Прибор на которым будем управлять	Сброс активации воспроизведения звуковых фрагментов для Рупор.
4	Пуск\Стоп речевого оповещения	1.Компьютер на котором находится устройство 2.Прибор - Прибор на которым будем управлять	Запускает все активированные на воспроизведения Рупор, подключенные к тому же интерфейсу, что и Прибор указанный в параметре №2
5	Выдать звуковое сообщение	1.Компьютер на котором находится устройство 2.Прибор - Прибор на который 3.Тип сообщения(Одиночный гудок, Двойной гудок и т.д.)	Управление звуковым спикером прибора
6	Сброс прибора	1.Компьютер на котором находится устройство 2.Прибор - Прибор на который 3.Вид сброса(Общий сброс, Сброс тревоги, Сброс неперданных событий)	Сброс прибора, необходим для перевода прибора в исходное состояние и активации тактик инициализации. Данная команда(Вид сброса=сброс тревоги) сбрасывает воспроизведение тревоги и активирует перевзятие зон и т.д.
7	Отправить сообщение на УО-4С	1.Компьютер на котором находится устройство 2.Прибор - Прибор на который	Транслирует сообщение возникшее на интерфейсе на приборы С2000-ИТ\УО-4С. Данный шаг работает, только при запуске сценария по событию. Запуск сценария в ручном режиме, по событию «Изменения состояния» и с дерева управления не приведет к отработке данного шага.
8	Управление режимом передачи	1.Компьютер на котором находится устройство 2.Прибор - Прибор на который, только С2000-ИТ	Управление режимом передачи для С2000ИТ, см. документацию на прибор.

»

4.2.Узел «Считыватель»

Номер	Наименование	Параметры	Описание
1	Заблокировать считыватель	1.Компьютер на котором находится считыватель 2.Считыватель	Блокирует доступ по направлению, который контролирует указанный считыватель. Блокируется как доступ по считывателю, так и по кнопке «Выход». Работает с приборами С2000-4\С2000-2

2	Открыть доступ	1.Компьютер на котором находится считыватель 2.Считыватель	Переводит прибор в состояние «Открыт доступ»(доступ разблокирован), по направлению контролируемого считывателем. Работает с приборами C2000-4\C2000-2\C2000 BIOAccess
3	Восстановление доступа	1.Компьютер на котором находится считыватель 2.Считыватель	Активирует-«включает» доступ на приборах, если прибор был в состоянии «Доступ открыт» или «Доступ заблокирован». Доступ активируется по направлению за которое отвечает указанный считыватель. Восстанавливается доступ как по считывателю, так и по кнопке «Выход». Работает с приборами C2000-4\C2000-2
4	Предоставить доступ	1.Компьютер на котором находится считыватель 2.Считыватель	Предоставить доступ по направлению считывателя Работает с приборами C2000-4\C2000-2\C2000 BIOAccess
5	Заблокировать кнопку «Выход»	1.Компьютер на котором находится считыватель 2.Считыватель	Блокировка доступа по кнопке «Выход» по направлению считывателя Работает с приборами C2000-4\C2000-2\C2000 BIOAccess
6	Разблокировать кнопку «Выход»	1.Компьютер на котором находится считыватель 2.Считыватель	Разблокировка доступа по кнопке «Выход» по направлению считывателя Работает с приборами C2000-4\C2000-2\C2000 BIOAccess
7	Заблокировать считыватель	1.Компьютер на котором находится считыватель 2.Считыватель	Блокировка доступа по считывателю. Кнопка Выход НЕ БЛОКИРУЕТСЯ. Работает с приборами C2000-4\C2000-2\C2000 BIOAccess
8	Разблокировать считыватель	1.Компьютер на котором находится считыватель 2.Считыватель	Разблокировка доступа по считывателю. Кнопка Выход НЕ БЛОКИРУЕТСЯ. Работает с приборами C2000-4\C2000-2\C2000 BIOAccess

4.3.Узел «Шлейф»

Номер	Наименование	Параметры	Описание
1	Взять на охрану ШС	1.Компьютер на котором находится шлейф 2.Шлейф на данном компьютере	Взять под охрану шлейф-сигнализации\зону. Работает для всех приборов ИСО Орион.
2	Снять с охраны ШС	1.Компьютер на котором находится шлейф 2.Шлейф на данном компьютере	Снять с охраны шлейф-сигнализации\зоны. Работает для всех приборов ИСО Орион.
3	Взять шлейф с задержкой	11.Компьютер на котором находится шлейф 2.Шлейф на данном компьютере 3.Задержка(в секундах)	Взять шлейф\зону под охрану с указанной задержкой, необходимо использовать данный шаг, для отработки тактики «Входного шлейфа».
4	Включение автоматики	1.Компьютер на котором находится шлейф	Включение пожарной автоматики пожаротушения.Работает только

		2.Шлейф на данном компьютере	для зон контролирующих состояние автоматики, для приборов С2000-АСПТ\Поток-3Н.
5	Выключение автоматики	1.Компьютер на котором находится шлейф 2.Шлейф на данном компьютере	Выключение пожарной автоматики пожаротушения.Работает только для зон контролирующих состояние автоматики, для приборов С2000-АСПТ\Поток-3Н.
6	Сброс тревоги	1.Компьютер на котором находится шлейф 2.Шлейф	После отсылки данной команды, прибор отрабатывает тактику сброса тревоги, как правило выключает тревожную сигнализацию и индикацию на приборе и пытается снять с охраны зону.
7	Сброс пожара	1.Компьютер на котором находится шлейф 2.Шлейф	После отсылки данной команды, прибор отрабатывает тактику сброса тревоги, как правило выключает тревожную сигнализацию и индикацию на приборе и пытается взять под охрану зону.

4.4.Узел «Камера»

Номер	Наименование	Параметры	Описание
1	Взять камеру под охрану	1.Компьютер на котором находится камера 2.Камера	Переводит состояние камеры в состояние «взята под охрану»
2	Снять камеру с охраны	1.Компьютер на котором находится камера 2.Камера	Снять с охраны камеру
3	Начать запись	1.Компьютер на котором находится камера 2.Камера	Включает запись по камере.
4	Выключить запись	1.Компьютер на котором находится камера 2.Камера	Выключение записи по камере
5	Включить детектор движения	1.Компьютер на котором находится камера 2.Камера	Включить детектор движения по камере(активировать либо аппаратный либо программный детектор движения)
6	Выключить детектор движения	1.Компьютер на котором находится камера 2.Камера	Выключить детектор движения по камере(выключить либо аппаратный либо программный детектор движения)
7	Повернуть	1.Компьютер на котором находится камера 2.Камера 3.Позиция, номер позиции камеры	Повернуть камеру в заранее запрограммируемую позицию камеры.
8	Показать камеру	1.Компьютер на котором находится камера 2.Камера	Вывести окно с камерой на экран компьютера
9	Скрыть камеру	1.Компьютер на котором	Скрыть окно с камерой с рабочего

		находится камера 2.Камера	экрана
--	--	------------------------------	--------

4.5.Узел «Выход

Номер	Наименование	Параметры	Описание
1	Команда Возврат в исходное состояние	1.Компьютер на котором находится выход 2.Реле\Управляемый выход	Возврат реле в исходное состояние, т.е. состояние заданное в конфигурации прибора.
2	Команда Включить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления	Включить-замкнуть реле.
3	Команда Выключить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления	Выключить-разомкнуть реле.
4	Команда Включить на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Время управления	Включить на время реле
5	Команда Выключить на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Время управления	Выключить на время реле
6	Команда Мигать из состояния выключить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Номер маски мигания(с 1-64)	Мигать реле из состояния выключено по указанной маски мигания
7	Команда Мигать из состояния включить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Номер маски мигания(с 1-64)	Мигать реле из состояния включено по указанной маски мигания
8	Команда Мигать из состояния выключить на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Время управления 5.Номер маски мигания(с 1-64)	Мигать реле из состояния выключено по указанной маски мигания и указанное время
9	Команда Мигать из состояния включено на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход 3.Задержка управления 4.Время управления 5.Номер маски мигания(с	Мигать реле из состояния включено по указанной маски мигания и указанное время

		1-64)	
9	Тактика 1-Включить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Включить, если есть «Тревога», «Пожар» или «Пожар 2»; иначе выключить. Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.
10	Тактика2-Выключить	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Выключить, если есть «Тревога», «Пожар» или «Пожар 2»; иначе включить. Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.
11	Тактика3-Включить на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Включить на заданное время при появлении в разделе состояния «Тревога», «Пожар» или «Пожар 2»; выключить по завершении времени управления или если нет зон в состоянии «Тревога», «Пожар», «Пожар 2» Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.
12	Тактика4-Выключить на время	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Выключить на заданное время при появлении в разделе состояния «Тревога», «Пожар» или «Пожар 2»; включить по завершении времени управления или если нет зон в состоянии «Тревога», «Пожар», «Пожар 2» Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.
13	Тактика5- «Мигать из состояния ВЫКЛЮЧЕНО»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Включить в прерывистом режиме (0,5 с включено, 0,5 с выключено), если есть «Тревога», «Пожар» или «Пожар 2»; иначе выключить.

			<p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
14	Тактика6- «Мигать из состояния ВКЛЮЧЕНО»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить в прерывистом режиме (0,5 с включено, 0,5 с выключено), если есть «Тревога», «Пожар» или «Пожар 2»;</p> <p>иначе включить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
15	Тактика7-«Мигать из состояния ВЫКЛЮЧЕНО на время»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить на заданное время в прерывистом режиме (0,5 с включено, 0,5 с выключено) при появлении в разделе состояния «Тревога», «Пожар» или «Пожар 2»;</p> <p>выключить по завершении времени управления или если нет зон в состоянии «Тревога», «Пожар», «Пожар 2».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика8- «Мигать из состояния ВКЛЮЧЕНО на время»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить на заданное время в прерывистом режиме (0,5 с включено, 0,5 с выключено) при появлении в разделе состояния «Тревога», «Пожар» или «Пожар 2»;</p> <p>включить по завершении времени управления или если нет зон в состоянии «Тревога», «Пожар», «Пожар 2».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика9- «ЛАМПА»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый</p>	<p>Если есть «Пожар 2», «Пожар», «Внимание», «Тревога», «Тревога входа»</p>

		выход	<p>или «Неудачное взятие», то мигать (0,5 с включено, 0,5 с выключено); если есть какая-либо неисправность1), то мигать (0,25 с включено, 1,75 с выключено); если есть взятый на охрану ШС, то включить; иначе выключить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика10- «ПЦН»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Если хотя бы одна зона связанных с выходом разделов в находится в тревожном состоянии («Пожар 2» , «Пожар», «Внимание», «Тревога», «Тревога входа»), либо есть неисправность ШС («Обрыв ШС», «Короткое замыкание ШС», «Неисправность», «ШС отключен», «Ошибка параметров зоны»), неисправность выхода или исполнительного устройства («Обрыв цепи выхода», «Короткое замыкание цепи выхода», «Выход отключен», «Некорректное состояние исполнительного устройства», «Отказ исполнительного устройства»), неисправность прибора («Вскрытие корпуса», «Короткое замыкание ДПЛС», «Авария ДПЛС»), либо есть зона в состоянии «Неудачное взятие» или «Снят», то выключить (разомкнуть); иначе включить (замкнуть).</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика11- «АСПТ»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить с заданной задержкой на заданное время, если в разделе появилось состояние «Пожар 2» (в состоянии «Пожар» перешел ручной пожарный извещатель или два и более автоматических пожарных извещателя) и при этом нет условий, блокирующих включение: «Технологический ШС нарушен»,</p>

			<p>«Выход отключен», «Обрыв цепи выхода», «Короткое замыкание цепи выхода». Появление блокирующего условия выключит выход. Также выход выключится по истечении заданного времени управления. Выход будет снова включен с задержкой после устранения блокировки, если сохранится состояние «Пожар 2»</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика12- «СИРЕНА»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если появилось состояние «Пожар» или «Пожар 2», то мигать заданное время (1,5 с включено, 0,5 с выключено); если «Внимание», то мигать заданное время (0,5 с включено, 1,5 с выключено); если «Тревога», то включить на заданное время; выключить по истечении заданного времени управления или если нет зон в состоянии «Пожар», «Пожар 2», «Внимание», «Тревога».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика13- «Пожарный ПЦН»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если есть «Пожар 2», «Пожар» или «Внимание», то включить (замкнуть); иначе выключить (разомкнуть).</p>
	Тактика14- «Выход НЕИСПРАВНОСТЬ»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если есть неисправность1), «Снят» или «Неудачное взятие», то выключить (разомкнуть); иначе включить (замкнуть).</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика15- «Пожарная ЛАМПА»	1.Компьютер- на котором находится выход	<p>Если есть «Пожар 2», «Пожар», «Внимание», «Тревога», «Тревога</p>

		2.Реле\Управляемый выход	<p>входа» или «Неудачное взятие», то мигать (0,5 с включено, 0,5 с выключено); если есть неисправность¹), то мигать (0,25 с включено, 1,75 с выключено); если есть зоны на охране («Взят») и нет зон, снятых с охраны («Снят»), то включить; иначе выключить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика16- «Старая тактика ПЦН»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если хотя бы одна зона связанных с выходом разделов находится в тревожном состоянии («Пожар 2» , «Пожар», «Внимание», «Тревога», «Тревога входа»), либо есть неисправность ШС («Обрыв ШС», «Короткое замыкание ШС», «Неисправность», «ШС отключен», «Ошибка параметров зоны»), неисправность выхода или исполнительного устройства («Обрыв цепи выхода», «Короткое замыкание цепи выхода», «Выход отключен», «Некорректное состояние исполнительного устройства», «Отказ исполнительного устройства»), неисправность прибора («Вскрытие корпуса», «Короткое замыкание ДПЛС», «Авария ДПЛС»), либо есть зона в состоянии «Неудачное взятие», то выключить (разомкнуть); иначе включить (замкнуть).</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика17- «Включить на время перед взятием»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если в разделе появилось состояние «Задержка взятия», то включить на заданное время; выключить по завершении времени управления или если нет зон в состоянии «Задержка взятия».</p>

			<p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	<p>Тактика18- «Выключить на время перед взятием»</p>	<p>1.Компьютер- на котором находится выход 2.Реле\Управляемый выход</p>	<p>Если в разделе появилось состояние «Задержка взятия», то выключить на заданное время; включить по завершении времени управления или если нет зон в состоянии «Задержка взятия».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	<p>Тактика19- «Включить на время при взятии»</p>	<p>1.Компьютер- на котором находится выход 2.Реле\Управляемый выход</p>	<p>Если в разделе появилось состояние «Взят» (если хотя бы одна зона раздела поставлена на охрану), то включить на заданное время; выключить по завершении времени управления или если нет зон в состоянии «Взят».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	<p>Тактика20- «Выключить на время при взятии»</p>	<p>1.Компьютер- на котором находится выход 2.Реле\Управляемый выход</p>	<p>Если в разделе появилось состояние «Взят» (если хотя бы одна зона раздела была поставлена на охрану), то выключить на заданное время; включить по завершении времени управления или если нет зон в состоянии «Взят».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	<p>Тактика21- «Включить на время при снятии»</p>	<p>1.Компьютер- на котором находится выход 2.Реле\Управляемый выход</p>	<p>Если в разделе появилось состояние «Снят» (если хотя бы одна зона раздела была снята с охраны), то включить на заданное время; выключить по завершении</p>

			<p>времени управления или если нет зон в состоянии «Снят».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика22- «Выключить на время при снятии»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если в разделе появилось состояние «Снят» (если хотя бы одна зона раздела была снята с охраны), то выключить на заданное время; включить по завершении времени управления или если нет зон в состоянии «Снят».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика23- «Включить на время при невзятии»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если в разделе появилось состояние «Неудачное взятие», то включить на заданное время; выключить по завершении времени управления или если нет зон в состоянии «Неудачное взятие».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика24- «Выключить на время при невзятии»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если в разделе появилось состояние «Неудачное взятие», то выключить на заданное время; включить по завершении времени управления или если нет зон в состоянии «Неудачное взятие».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика25- «Включить на время при нарушении технологического	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если в разделе появилось состояние «Технологический ШС нарушен», то включить на заданное время;</p>

	ШС»		<p>выключить по завершении времени управления или если нет нарушенных технологических ШС.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика26- «Выключить на время при нарушении технологического ШС»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Если в разделе появилось состояние «Технологический ШС нарушен», то выключить на заданное время; включить по завершении времени управления или если нет нарушенных технологических ШС.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика27- «Включить при снятии»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Если есть снятые с охраны зоны, то включить; иначе выключить</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика28- «Выключить при снятии»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Если есть снятые с охраны зоны, то выключить; иначе включить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика29- «Включить при взятии»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Если есть зоны на охране, то включить; иначе выключить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика30- «Выключить при взятии»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый</p>	<p>Если есть зоны на охране, то выключить; иначе включить.</p>

		выход	<p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика31- «Включить при нарушении технологического ШС»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если есть нарушенная технологическая зона, то включить; иначе выключить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика32- «Выключить при нарушении технологического ШС»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Если есть нарушенная технологическая зона, то выключить; иначе включить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика33- «АСПТ-1»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время, если в разделе появилось состояние «Пожар» (пожарная зона перешла в состояние «Пожар» или «Пожар 2») и нет условий, блокирующих включение: «Технологический ШС нарушен», «Выход отключен», «Обрыв цепи выхода» или «Короткое замыкание цепи выхода».</p> <p>Появление блокирующего условия выключит выход. Выход будет снова включен с задержкой после устранения блокировки, если сохранится состояние «Пожар».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика34- «АСПТ-А»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить с заданной задержкой на заданное время, если в разделе появилось состояние «Пожар 2» (в состоянии «Пожар» перешел ручной</p>

			<p>пожарный извещатель или два и более автоматических пожарных извещателя) и нет условий, блокирующих включение: «Технологический ШС нарушен», «Выход отключен», «Обрыв цепи выхода» или «Короткое замыкание цепи выхода». Появление блокирующего условия приведет к выключению выхода. После устранения блокировки выход останется выключенным.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика35- «АСПТ-А1»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить на заданное время, если в разделе появилось состояние «Пожар» (пожарная зона перешла в состояние «Пожар» или «Пожар 2») и нет условия блокировки: («Технологический ШС нарушен», «Выход отключен», «Обрыв цепи выхода» или «Короткое замыкание цепи выхода»). Появление блокирующего условия приведет к выключению выхода. После устранения блокировки выход останется выключенным.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика36- «Включить при повышении уровня»	<p>1.Компьютер- на котором находится выход</p> <p>2.Реле\Управляемый выход</p>	<p>Включить на заданное время при повышении температуры выше «порога повышения температуры» (состояние «Повышенная температура»), при превышении заданного «верхнего» уровня влажности или уровня воды в резервуаре (состояние «Повышение уровня»). Иначе выключить.</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из</p>

			списка разделов на управление, указанных для реле, см.настройки управления.
	Тактика37- «Включить при понижении уровня»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время при понижении температуры ниже «порога понижения температуры» (состояние «Пониженная температура»), при понижении влажности или уровня воды в резервуаре ниже заданного «нижнего» порога (состояние «Понижение уровня»). Иначе выключить</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика38- «Включить при задержке пуска АУП»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время, если идет задержка перед выдачей импульса запуска автоматической установки пожаротушения (АУП) (в разделе появилось состояние «Задержка запуска АУП»); выключить по завершении времени управления или если нет состояния «Задержка запуска АУП».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика39- «Включить при пуске АУП»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время, если выдан импульс запуска АУП (в разделе появилось состояние «Пуск АУП»); выключить по завершении времени управления или если нет состояния «Пуск АУП».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика40- «Включить при тушении»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время при подтвержденном пуске АУП (в разделе появилось состояние «Тушение»); выключить по завершении времени управления или если нет состояния</p>

			<p>«Тушение».</p> <p>41 «Включить при неудачном пуске АУП»</p> <p>Включить на за</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика41- «Включить при включении автоматики»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время, если в разделе появилось состояние «Автоматика включена»;</p> <p>выключить по завершении времени управления или если нет состояния «Автоматика включена»</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика42- «Выключить при включении автоматики»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Выключить на заданное время, если в разделе появилось состояние «Автоматика включена»;</p> <p>включить по завершении времени управления или если нет состояния «Автоматика включена».</p>
	Тактика43- «Включить при выключении автоматики»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Включить на заданное время, если в разделе появилось состояние «Автоматика выключена»;</p> <p>выключить по завершении времени управления или если нет состояния «Автоматика выключена».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.</p>
	Тактика44- «Выключить при выключении автоматики»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	<p>Выключить на заданное время, если в разделе появилось состояние «Автоматика выключена»;</p> <p>включить по завершении времени управления или если нет состояния «Автоматика выключена».</p> <p>Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление,</p>

			указанных для реле, см.настройки управления.
	Тактика45- «Включить, если исполнительное устройство в рабочем состоянии»	1.Компьютер- на котором находится выход 2.Реле\Управляемый выход	Включить на заданное время, если исполнительное устройство перешло в рабочее состояние; выключить по завершении времени управления или если нет исполнительных устройств в рабочем состоянии. Данную тактику необходимо запускать, только по спец событию у раздела «Изменение состояния». Список разделов берется из списка разделов на управление, указанных для реле, см.настройки управления.

4.6.Узел «Группа Разделов»

Номер	Наименование	Параметры	Описание
1	Взять группу разделов	1.Компьютер на котором находится Группа разделов 2.Группа разделов	Взять под охрану группу разделов
2	Снять группу разделов	1.Компьютер на котором находится Группа разделов 2.Группа разделов	Снять с охраны группу разделов
3	Взять гр.разд. по уходу последнего из заданной зоны	1.Зона доступа 2.Группа разделов 3.Компьютер на котором находится Группа разделов	Взять под охрану группу разделов при выходе последнего сотрудника из указанной зоны доступа. Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
4	Снять гр.разд.по приходу первого в заданную зону	1.Зона доступа 2.Группа разделов 3.Компьютер на котором находится Группа разделов	Снять с охраны группу разделов при входе первого сотрудника из указанной зоны доступа. Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
5	Взять гр.разд. после прохода последнего с уровнем доступа из зоны доступа	1.Компьютер на котором находится Группа разделов 2.Группа разделов 3.Зона доступа 4.Уровень доступа	Взять под охрану группу разделов при выходе последнего сотрудника из указанной зоне доступа, сотрудника определенного уровня доступа Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
6	Снять гр.разд. после прохода первого с уровнем доступа в зону доступа	1.Компьютер на котором находится Группа разделов 2.Группа разделов 3.Зона доступа	Снять с охраны группу разделов при входе первого сотрудника из указанной зоне доступа определенного уровня

		4.Уровень доступа	доступа(параметр 4). Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
7	Взять гр.разд. после прохода последнего с подразделением из зоны доступа	1.Компьютер на котором находится Группа разделов 2.Группа разделов 3.Зона доступа 4.Подразделение	Взять под охрану группу разделов при выходе последнего сотрудника из указанной зоне доступа, сотрудника определенного подразделения(параметр 4) Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
8	Снять гр.разд. после прохода первого с подразделением в зону доступа	1.Компьютер на котором находится Группа разделов 2.Группа разделов 3.Зона доступа 4.Подразделение	Снять с охраны группу разделов при входе первого сотрудника из указанной зоне доступа определенного подразделения(параметр 4). Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
9	Взять\Снять группу событий	1.Компьютер на котором находится Группа разделов 2.Группа разделов	Если группа разделов была в состоянии «Взят» она будет снята, в противном случае будет взята.

4.7.Узел «Раздел»

Номер	Наименование	Параметры	Описание
1	Пуск АСПТ	1.Компьютер на котором находится раздел 2.Раздел	Пуск АСПТ по разделу.Данная команда выполняется если в разделе имеется зона контролирующая пуск АСПТ(только для приборов С2000-АСПТ\Поток-3Н).
2	Включение автоматики	1.Компьютер на котором находится раздел 2.Раздел	Включить автоматику по разделу. Данная команда выполняется если в разделе имеется зона контролирующая пуск АСПТ(только для приборов С2000-АСПТ\Поток-3Н).
3	Выключение автоматики	1.Компьютер на котором находится раздел 2.Раздел	Выключить автоматику по разделу. Данная команда выполняется если в разделе имеется зона контролирующая пуск АСПТ(только для приборов С2000-АСПТ\Поток-3Н).
4	Сброс тревоги	1.Компьютер на котором находится раздел 2.Раздел	Сброс охранной тревоги в разделе
5	Сброс пожара	1.Компьютер на котором находится раздел 2.Раздел	Сброс пожарной тревоги в разделе
6	Взять раздел	1.Компьютер на котором находится раздел 2.Раздел	Взять под охрану раздел
7	Снять раздел	1.Компьютер на котором находится раздел 2.Раздел	Снять с охраны раздел

8	Взятие разделов привязанных к рабочему месту	1.Компьютер на котором находится раздел	Взятие всех разделов на компьютер
9	Снятие разделов привязанных к рабочему месту	1.Компьютер на котором находится раздел	Снятие разделов привязанных к рабочему месту
10	Взять раздел по уходу последнего из заданной зоны	1.Зона доступа 2.Раздел 3.Компьютер на котором находится раздел	Взять под охрану раздел при выходе последнего сотрудника из указанной зоны доступа. Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
11	Снять раздел .по приходу первого в заданную зону	1.Зона доступа 2.Раздел 3.Компьютер на котором находится раздел	Снять с охраны раздел при входе первого сотрудника из указанной зоны доступа. Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
12	Взять раздел после прохода последнего с уровнем доступа из зоны доступа	1.Компьютер на котором находится раздел 2.Раздел 3.Зона доступа 4.Уровень доступа	Взять под охрану раздел при выходе последнего сотрудника из указанной зоны доступа, сотрудника определенного уровня доступа Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
13	Снять раздел. после прохода первого с уровнем доступа в зону доступа	1.Компьютер на котором находится раздел 2.Раздел 3.Зона доступа 4.Уровень доступа	Снять с охраны раздел при входе первого сотрудника из указанной зоны доступа определенного уровня доступа(параметр 4). Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
14	Взять раздел. после прохода последнего с подразделением из зоны доступа	1.Компьютер на котором находится раздел 2.Группа разделов 3.Зона доступа 4.Подразделение	Взять под охрану раздел при выходе последнего сотрудника из указанной зоны доступа, сотрудника определенного подразделения(параметр 4) Сценарий необходимо запускать по событию «Проход» для зоны доступа примыкающей к Зоне в которой считаем сотрудников(параметр 1).
15	Снять раздел. после прохода первого с подразделением в зону доступа	1.Компьютер на котором находится раздел 2.раздел 3.Зона доступа 4.Подразделение	Снять с охраны раздел при входе первого сотрудника из указанной зоны доступа определенного подразделения(параметр 4). Сценарий необходимо запускать по событию «Проход» для зоны указанной в параметре 1
16	Взять\Снять разлел	1.Компьютер на котором находится раздел	Если раздел была в состоянии «Взят» она будет снята, в

		2.раздел	противном случае будет взята.

4.8.Узел «Ядро»

Номер	Наименование	Параметры	Описание
1	Выдать сообщение во всплывающем окне	1.Компьютер на котором запускается ядро 2.Текст сообщения	Выдает сообщение во всплывающем окне.
2	Выдать текстовое сообщение в ядро	1.Компьютер на котором запускается ядро 2.Текст сообщения	Записать сообщение в ядре опроса на вкладке Сценарии. Данный шаг создан для отладки сложных сценариев управления.
3	Пауза	1.Пауза(в секундах)	Показывает окно с уменьшающим отсчетом секунд(параметр 1)
4	Пауза(задержка)	1.Пауза(в секундах)	Данный шаг генерирует паузы в сценарии. Необходимо использовать при генерации задержки внутри сценария
5	Запрос оператору	1.Компьютер на котором запускается ядро 2.Текст запроса 3.Время запроса(в секундах) 4.Запустить сценарий если ответ, да, ссылка на другой сценарий 5.Запустить сценарий если ответ, Нет, ссылка на другой сценарий	Показывает окно с вопросом, если ответ да, то запускается сценарий указанный в параметре 4, иначе параметр 4. Ожидание ответа длится указанное в Параметре №3. Если пользователь не осуществляет выбор по умолчанию отработывает сценарий указанный в параметре №4.
6	Воспроизведение звукового файла	1.Компьютер на котором запускается ядро 2.Путь и имя файла, которое будет воспроизведено.	Проигрывание звукового файла, поддерживает только wav формат. Необходимо указывать полный путь к файлу.
8	Показать камеру	1.Компьютер на котором запускается ядро опроса 2.Камера	Показывает камеру в окне на компьютере с ядром опроса. Камера не обязательно должна контролироваться видеодрайвером, который запускается на данном рабочем месте.
9	Скрыть камеру	1.Компьютер на котором запускается ядро опроса 2.Камера	Скрыть камеру в окне на компьютере с ядром опроса. Камера не обязательно должна контролироваться видеодрайвером, который запускается на данном рабочем месте.
10	Запуск внешней программы	1.Компьютер на котором запускается ядро 2.Путь и имя файла, которое будет воспроизведено.	Запуск программы. В параметре 2, необходимо указать полный путь и имя исполнительного файла, также возможно указать параметры.
11	Отправка email с вводом текста	1.Компьютер на котором запускается ядро 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика,	Появляется окно, которое предлагает вести пользователю адрес на который будет выслано письмо,тему письма и текст. Внимание! Большинство почтовый серверов работают только с

		<p>через который отсылаем почту</p> <p>5.Пароль, пароль почтового сервера через который отсылаем почту</p> <p>6.Адрес отправителя, почтовый адрес через который шлем письмо</p> <p>7.Тип сообщение:Используем или не Используем SSL шифрование</p>	включенным SSL шифрованием.
12	Отправка email	<p>1.Компьютер на котором запускается ядро</p> <p>2.Порт почтового сервера</p> <p>3.SMTP-хост, имя почтового сервера</p> <p>4.Логин, почтового ящика, через который отсылаем почту</p> <p>5.Пароль, пароль почтового сервера через который отсылаем почту</p> <p>6.Адрес отправителя, почтовый адрес через который шлем письмо</p> <p>7.Адрес получателя, адрес на который высылаем письмо</p> <p>8.Тема письма</p> <p>9.Текст письма</p> <p>10.Тип сообщение:Используем или не Используем SSL шифрование</p>	<p>Отправка email, в качестве текста, возможно задать, любое однострочное текстовое значение. Внимание! Большинство почтовых серверов работают только с включенным SSL шифрованием</p> <p>Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.</p>
13	Отправка email по событию	<p>1.Компьютер на котором запускается ядро</p> <p>2.Порт почтового сервера</p> <p>3.SMTP-хост, имя почтового сервера</p> <p>4.Логин, почтового ящика, через который отсылаем почту</p> <p>5.Пароль, пароль почтового сервера через который отсылаем почту</p> <p>6.Адрес отправителя, почтовый адрес через который шлем письмо</p> <p>7.Адрес получателя, адрес на который высылаем письмо</p> <p>8.Тема письма</p> <p>9.Тип сообщение:Используем или не Используем SSL шифрование</p>	<p>Отправка email, где текст письма собирается по параметрам от события возникшем в системе. Данный шаг выполняется только по событию, и не выполняется по запуску из дерева управления, расписанию, горячей клавише. Большинство почтовых серверов работают только с включенным SSL шифрованием</p> <p>Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.</p>
14	Подключится к почтовому серверу	<p>1.Компьютер на котором запускается ядро</p> <p>2.Имя почтового сервера</p> <p>3.Логин, почтового ящика, через который отсылаем почту</p>	<p>Команда подключения к почтовому серверу, для дальнейшего удаленного управления, через него с помощью посылки электронных писем.</p>

		4.Пароль, пароль почтового сервера через который отсылаем почту 5.Порт почтового сервера 6.При вычитывании писем, удалять их из почтового сервера 7.Включение SSL шифрования. По умолчанию включено	
15	Принять почту	1.Компьютер на котором запускается ядро 2.Имя почтового сервера	Запуск чтения почты с почтового сервера
16	Разрешить принимать почту с почтового адреса	1.Компьютер на котором запускается ядро 2.Имя почтового сервера 3.адрес почты	Разрешает управление с почтового адреса указанного в параметре №3.
17	Запретить принимать почту с почтового адреса	1.Компьютер на котором запускается ядро 2.Имя почтового сервера 3.адрес почты	Запрещает управление с почтового адреса указанного в параметре №3.
18	Отключиться от почтового сервера	1.Компьютер на котором запускается ядро 2.Имя почтового сервера	Отключение от почтового сервера. Для принятия писем необходимо будет снова выполнить шаг «Подключиться к почтовому серверу»
19	Остановить прием почты	1.Компьютер на котором запускается ядро 2.Имя почтового сервера	Останавливает прием почты, не отключаясь от почтового сервера.
20	Воспроизвести пользовательское сообщение	1.Компьютер на котором запускается ядро 2.Сообщение(текстовое поле) 3.Количество повторов 4.Пауза между повторами	Воспроизведение текстового сообщения через синтезатор голоса, требует наличия на рабочем месте soundServer.
21	Воспроизвести сообщение возникшее на интерфейсе	1.Компьютер на котором запускается ядро 2.Количество повторов 3.Пауза между повторами	Воспроизведение сообщения собранного из события через синтезатор голоса. Данный шаг выполняется только по событию, и не выполняется по запуску из дерева управления, расписанию, горячей клавише.
22	Отключить звуковое оповещение	1.Компьютер на котором запускается ядро	Отключает воспроизведение звуковых сообщений. Внимание, любые звуковые сообщения воспроизводятся не будут!
23	Включить звуковое оповещение	1.Компьютер на котором запускается ядро	Включает воспроизведение звуковых сообщений.
24	Активировать рупор	1.Компьютер на котором запускается ядро находится устройство 2.Прибор - Прибор на который высылает сообщение 3.Задержка управления, в секундах. Задержка	Активируется воспроизведение звуковых сообщений, записанных в Рупор. Активация не означает воспроизведение фрагмента, прибор переводится в состояние готовности воспроизведения. Для воспроизведения фрагментов необходимо осуществить запуск

		воспроизведения звукового фрагмента 4.Время воспроизведения, звукового фрагмента, в секундах 5.Номер звукового фрагмента	воспроизведения, с помощью команды «Запустить РУПОР».
25	Запустить РУПОР	1.Компьютер на котором запускается ядро 2.Прибор - Прибор на которым будем управлять	Сброс активации воспроизведения звуковых фрагментов для Рупор.
26	Пуск\Стоп речевого оповещения	1.Компьютер на котором запускается ядро 2.Прибор - Прибор на которым будем управлять	Запускает все активированные на воспроизведения Рупор, подключенные к тому же интерфейсу, что и Прибор указанный в параметре №2
27	Ограничить выполнение	1.Компьютер на котором запускается ядро	Шаги сценария указанного ниже будут выполняться, только на компьютере указанном в параметре №1

4.9.Узел «Сценарий»

Номер	Наименование	Параметры	Описание
1	Запустить сценарий	1.Сценарий 2.Задержка(в секундах)	Запускает сценарий управления
2	Запуск сценария по выбору	1.Компьютер 2.Сценарий который будет запущен если нажали Да 3. Сценарий который будет запущен если нажали Нет	Записать сообщение в ядре опроса на вкладке Сценарии. Данный шаг создан для отладки сложных сценариев управления.
3	Остановить сценарий	1.Сценарий	Останавливает выполнение сценарии. Внимание, сценарий может быть остановлен, только при условии, что есть команды ожидающие ввод данных или используется команда Пауза или Задержка.
4	Заблокировать сценарий	1.Сценарий	Сценарий помечается, как заблокированный и при последующем его вызове не будет выполняться
5	Разблокировать сценарий	1.Сценарий	Сценарий помечается, как разблокированный,.

4.10.Узел «Монитор»

Номер	Наименование	Параметры	Описание
1	Запрос оператору	1.Компьютер 2.Текст запроса 3.Время запроса(в секундах) 4.Запустить сценарий если ответ, да, ссылка на другой сценарий 5.Запустить сценарий если ответ, Нет, ссылка на другой сценарий	Показывает окно с вопросом, если ответ да, то запускается сценарий указанный в параметре 4, иначе параметр 4. Ожидание ответа длится указанное в Параметре №3. Если пользователь не осуществляет выбор по умолчанию отработывает сценарий указанный в параметре №4.
2	Запуск внешней программы	1.Компьютер 2.Путь и имя файла, которое будет воспроизведено.	Запуск программы. В параметре 2, необходимо указать полный путь и имя исполнительного файла, также

			возможно указать параметры.
3	Воспроизведение звукового файла	1.Компьютер на котором запускается ядро 2.Путь и имя файла, которое будет воспроизведено.	Проигрывание звукового файла, поддерживает только wav формат. Необходимо указывать полный путь к файлу.
4	Выдача сообщения на монитор	1.Компьютер на котором запускается ядро 2.Текст сообщения	Выдает сообщение во всплывающем окне.
5	Отправка email	1.Компьютер 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика, через который отсылаем почту 5.Пароль, пароль почтового сервера через который отсылаем почту 6.Адрес отправителя, почтовый адрес через который шлем письмо 7.Адрес получателя, адрес на который высылаем письмо 8.Тема письма 9.Текст письма 10.Тип сообщение:Используем или не Используем SSL шифрование	Отправка email, в качестве текста, возможно задать, любое однострочное текстовое значение. Внимание! Большинство почтовых серверов работают только с включенным SSL шифрованием Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.
6	Отправка email по событию	1.Компьютер 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика, через который отсылаем почту 5.Пароль, пароль почтового сервера через который отсылаем почту 6.Адрес отправителя, почтовый адрес через который шлем письмо 7.Адрес получателя, адрес на который высылаем письмо 8.Тема письма 9.Тип сообщение:Используем или не Используем SSL шифрование	Отправка email, где текст письма собирается по параметрам от события возникшем в системе. Данный шаг выполняется только по событию, и не выполняется по запуску из дерева управления, расписанию, горячей клавише. Большинство почтовых серверов работают только с включенным SSL шифрованием Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.
7	Воспроизвести пользовательское сообщение	1.Компьютер 2.Сообщение(текстовое поле) 3.Количество повторов 4.Пауза между повторами	Воспроизведение текстового сообщения через синтезатор голоса, требует наличия на рабочем месте soundServer.
8	Воспроизвести сообщение возникшее на интерфейсе	1.Компьютер 2.Количество повторов 3.Пауза между повторами	Воспроизведение сообщения собранного из события через синтезатор голоса. Данный шаг выполняется только по событию, и не выполняется по

			запуску из дерева управления, расписанию, горячей клавише.
9	Записать событие в протокол	1.Компьютер 2.Текст	Записывает текстовое сообщение в протокол, не более 40 символов.
10	Отключить звуковое оповещение	1.Компьютер	Отключает воспроизведение звуковых сообщений. Внимание, любые звуковые сообщения воспроизводятся не будут!
11	Включить звуковое оповещение	1.Компьютер	Включает воспроизведение звуковых сообщений.
12	Запуск хранителя экрана	1.Компьютер	Запуск хранителя экран
13	Запуск окна со сменой дежурства	1.Компьютер	Активация процедуры смены дежурства
14	Вывод отчета за смену	1.Компьютер	Вывод отчета по тревогам за текущую смену
15	Отправка email с вводом текста	1.Компьютер 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика, через который отсылаем почту 5.Пароль, пароль почтового сервера через который отсылаем почту 6.Адрес отправителя, почтовый адрес через который шлем письмо 7.Тип сообщение:Используем или не Используем SSI шифрование	Появляется окно, которое предлагает вести пользователю адрес на который будет выслано письмо,тему письма и текст. Внимание! Большинство почтовый серверов работают только с включенным SSL шифрованием.

4.11.Узел «Оболочка»

Номер	Наименование	Параметры	Описание
1	Показать камеру	1.Компьютер 2.Камера	Показывает камеру в окне на компьютере с ядром опроса. Камера не обязательно должна контролироваться видеодрайвером, который запускается на данном рабочем месте.
2	Скрыть камеру	1.Компьютер 2.Камера	Скрыть камеру в окне на компьютере с ядром опроса. Камера не обязательно должна контролироваться видеодрайвером, который запускается на данном рабочем месте.
3	Запрос оператору	1.Компьютер 2.Текст запроса 3.Время запроса(в секундах) 4.Запустить сценарий если ответ, да, ссылка на другой сценарий 5.Запустить сценарий если ответ, Нет, ссылка на другой сценарий	Показывает окно с вопросом, если ответ да, то запускается сценарий указанный в параметре 4, иначе параметр 4. Ожидание ответа длится указанное в Параметре №3. Если пользователь не осуществляет выбор по умолчанию обрабатывает сценарий указанный в параметре №4.

4	Воспроизведение звукового файла	1.Компьютер на котором запускается ядро 2.Путь и имя файла, которое будет воспроизведено.	Проигрывание звукового файла, поддерживает только wav формат. Необходимо указывать полный путь к файлу.
5	Запуск внешней программы	1.Компьютер 2.Путь и имя файла, которое будет воспроизведено.	Запуск программы. В параметре 2, необходимо указать полный путь и имя исполнительного файла, также возможно указать параметры.
6	Выдача сообщения на монитор	1.Компьютер на котором запускается ядро 2.Текст сообщения	Выдает сообщение во всплывающем окне.
7	Отправка email по событию	1.Компьютер 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика, через который отсылаем почту 5.Пароль, пароль почтового сервера через который отсылаем почту 6.Адрес отправителя, почтовый адрес через который шлем письмо 7.Адрес получателя, адрес на который высылаем письмо 8.Тема письма 9.Тип сообщение:Используем или не Используем SSL шифрование	Отправка email, где текст письма собирается по параметрам от события возникшем в системе. Данный шаг выполняется только по событию, и не выполняется по запуску из дерева управления, расписанию, горячей клавише. Большинство почтовых серверов работают только с включенным SSL шифрованием Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.
8	Отправка email	1.Компьютер 2.Порт почтового сервера 3.SMTP-хост, имя почтового сервера 4.Логин, почтового ящика, через который отсылаем почту 5.Пароль, пароль почтового сервера через который отсылаем почту 6.Адрес отправителя, почтовый адрес через который шлем письмо 7.Адрес получателя, адрес на который высылаем письмо 8.Тема письма 9.Текст письма 10.Тип сообщение:Используем или не Используем SSL шифрование	Отправка email, в качестве текста, возможно задать, любое однострочное текстовое значение. Внимание! Большинство почтовых серверов работают только с включенным SSL шифрованием Также на компьютере должна быть разрешена передача данных по smtp\pop3 протоколу.
9	Воспроизвести пользовательское сообщение	1.Компьютер 2.Сообщение(текстовое поле) 3.Количество повторов 4.Пауза между повторами	Воспроизведение текстового сообщения через синтезатор голоса, требует наличия на рабочем месте soundServer.
10	Воспроизвести сообщение	1.Компьютер 2.Количество повторов	Воспроизведение сообщения собранного из события через

	возникшее на интерфейсе	3.Пауза между повторами	синтезатор голоса. Данный шаг выполняется только по событию, и не выполняется по запуску из дерева управления, расписанию, горячей клавише.
11	Отключить звуковое оповещение	1.Компьютер	Отключает воспроизведение звуковых сообщений. Внимание, любые звуковые сообщения воспроизводятся не будут!
12	Включить звуковое оповещение	1.Компьютер	Включает воспроизведение звуковых сообщений.
13	Записать событие в протокол	1.Компьютер 2.Текст	Записывает текстовое сообщение в протокол, не более 40 символов.

4.12.Узел «Ключницы и пеналы»

Номер	Наименование	Параметры	Описание
1	Открыть пенал ключницы	1.Компьютер 2.Пенал	Открыть пенал, сделать его доступным

4.13.Узел «Сотрудники»

Номер	Наименование	Параметры	Описание
1	Предоставить доступ	1.Компьютер 2.Считыватель 3.Сотрудник	Предоставить именной доступ по заданному направлению, которое контролирует считыватель. Для контроллеров доступа С2000-2\С2000-4 от 2012 года выпуска и далее
2	Восстановить доступ	1.Компьютер 2.Считыватель 3.Сотрудник	Восстановить именной доступ по заданному направлению, которое контролирует считыватель. Для контроллеров доступа С2000-2\С2000-4 от 2012 года выпуска и далее. Команда восстанавливает доступ в случае если он был «Заблокирован» или «Открыт на свободный проход»
3	Открытие доступ	1.Компьютер 2.Считыватель 3.Сотрудник	Открыть именной доступ по заданному направлению, которое контролирует считыватель. Для контроллеров доступа С2000-2\С2000-4 от 2012 года выпуска и далее
4	Заблокировать считыватель	1.Компьютер 2.Считыватель 3.Сотрудник	Заблокировать считыватель от указанного сотрудника по заданному направлению, которое контролирует считыватель. Для контроллеров доступа С2000-2\С2000-4 от 2012 года выпуска и далее
5	Разблокировать считыватель	1.Компьютер 2.Считыватель 3.Сотрудник	Разблокировать считыватель от указанного сотрудника по заданному направлению, которое контролирует считыватель. Для контроллеров доступа С2000-2\С2000-4 от 2012 года выпуска и далее
6	Заблокировать ключи доступа	1.Компьютер 2.Сотрудник	Блокирует все ключи доступа у сотрудника. Ключ доступа

			блокируется в приборе, равнозначно установке значение ключа, ключ заблокирован.-
7	Разблокировать ключи доступа	1.Компьютер 2.Сотрудник	Разблокирует все ключи доступа у сотрудника. Ключ доступа блокируется в приборе, равнозначно установке значение ключа, ключ заблокирован.-

5. [Примеры сценариев.](#)

5.1. [Перечень примеров сценариев.](#)

1. *Пример сценария, запускаемого по событию в системе с несколькими Ядрами опроса; Или запускаемого любым способом в системе с одним Ядром опроса.*

Так как сценарий, запускаемый по событию в системе с несколькими Ядрами опроса, или запускаемый любым способом в системе с одним Ядром опроса, выполняется только в одном Ядре опроса, то команда объекту системы будет послана только одна, и нет необходимости производить какие-либо дополнительные проверки.

Рассмотрим пример, когда необходимо взять раздел на охрану (с номером 7).

Внимание! Примеры текстов простых сценариев также можно посмотреть в папке с установленным АРМ «Орион Про» в папке DOC \ SCRIPTS.

[Пример 1.](#)

2. *Пример простого сценария, запускаемого по горячей клавише, из дерева управления, или по расписанию в системе с несколькими Ядрами опроса.*

Так как сценарий запускаемый по горячей клавише, из дерева управления, или по расписанию, запускается сразу во всех Ядрах опроса, то в большинстве случаев необходимо разделять объекты, которыми производится управление, по принадлежности к конкретному Ядру опроса.

Другими словами, необходимо проверять, принадлежит ли объект, для которого посылаются команды, тому Ядру опроса, в котором выполняется сценарий.

Рассмотрим пример, когда необходимо блокировать доступ для одной точки доступа (с номером 3, и принадлежащей Ядру опроса с IP-адресом «192.168.11.5»). В тексте сценария производится проверка на принадлежность точки доступа Ядру опроса, в котором и запущен сценарий.

Данный сценарий запустится во всех Ядрах опроса, но команда на блокирование доступа будет произведена только в одном из них.

[Пример 2.](#)

3. *Пример сценария, запускаемого по горячей клавише, из дерева управления, или по расписанию в системе с несколькими Ядрами опроса.*

Исходные условия для сценария аналогичны условиям, описанным в предыдущем примере.

А теперь рассмотрим пример, когда необходимо блокировать доступ для всех точек доступа системы. В тексте сценария производится перемещение по всем Ядрам опроса, и выполняются действия только для тех объектов, которые принадлежат Ядру опроса, в котором и запущен сценарий.

Данный сценарий запустится во всех Ядрах опроса, но команды на блокирование доступа будут произведены в каждом Ядре опроса только для точек доступа, которые принадлежат каждому конкретному Ядру опроса.

[Пример 3.](#)

4. Отправка email с текстом «Привет мир!» через почтовый ящик orion@mail.ru на mymail@mail.ru, через порт с передачей данных с использованием SSL

[Пример 4.](#)

- 5 Трансляция события на УО-4С, скрипт запускается по событиям Контроля доступа.

[Пример 5.](#)

- 6 Ограничение выполнения сценария на компьютере с ID=1
Шаги с взятием шлейфов будут выполняться, только на компьютере с ID=1

[Пример 6.](#)

- 7 Сценарий блокирующий ключ на 1 час.

[Пример 7.](#)

8. Настройка запуска сценариев через Email. Настройка управление происходит в 3 шага.

1) Подключение к почтовому smtp серверу maul.ru smtp.mail.ru

2) Разрешение управления 111@bolid.ru, 222@bolid.ru, 333@gmail.com

3) активация вычитания почты, письма вычитываются с удалением.

[Пример 8.](#)

9. Воспроизвести звуковое сообщение в формате "*.wav" через soundServer.
Текст сообщения конвертируется в файл

[Пример 9.](#)

10. Оповещение через SoundServer.

[Пример 10.](#)

11. Запись сообщения в журнал событий

[Пример 11.](#)

5.2. [Тексты сценариев.](#)

[Пример 1.](#)

Сценарий

Переменные

Перем Разделы1;

Перем Раздел1;

НачалоСценария

Разделы1 = СоздатьОбъект("Разделы");

Если Не ПустоеЗначение(Разделы1) Тогда

Раздел1 = Разделы1.ПолучитьПоИндексу(7);

Если Не ПустоеЗначение(Раздел1) Тогда

Сообщить(Раздел1.Название(), 0);

Раздел1.Взять();

КонецЕсли;

КонецЕсли;

КонецСценария

[Пример 2.](#)

Сценарий

Переменные

Перем Компьютеры1;

Перем РабочееМесто1;

Перем Двери1;

Перем Дверь1;

Перем Считыватель1;

НачалоСценария

Компьютеры1 = СоздатьОбъект("Компьютеры");

Если Не ПустоеЗначение(Компьютеры1) Тогда

РабочееМесто1 = Компьютеры1.ПолучитьПоТипуИСетевомуАдресу("Ядро", "192.168.11.5");

Если Не ПустоеЗначение(РабочееМесто1) Тогда

Если РабочееМесто1.Локальное() == ИСТИНА Тогда // Требуемая проверка

```

Двери1 = РабочееМесто1.Двери();
Если Не ПустоеЗначение( Двери1 ) Тогда
    Дверь1 = Двери1.ПолучитьПоИндексу( 3 );
Если Не ПустоеЗначение( Дверь1 ) Тогда
    Сообщить( Дверь1.Название(), 0 );
    Считыватель1 = Дверь1.СчитывательНаВход();
    Если Не ПустоеЗначение( Считыватель1 ) Тогда
        Сообщить( " " + Считыватель1.Название(), 0 );
        Считыватель1.ЗаблокироватьДоступ();
    КонецЕсли;
    Считыватель1 = Дверь1.СчитывательНаВыход();
    Если Не ПустоеЗначение( Считыватель1 ) Тогда
        Сообщить( " " + Считыватель1.Название(), 0 );
        Считыватель1.ЗаблокироватьДоступ();
    КонецЕсли;
КонецЕсли;
КонецЕсли;
КонецЕсли;
КонецЕсли;
КонецСценария

```

Пример 3.

Сценарий

Переменные

```

Перем Компьютеры1;
Перем РабочееМесто1;
Перем Двери1;
Перем Считыватель1;
Перем Х;
Перем У;

```

НачалоСценария

```

Компьютеры1 = СоздатьОбъект( "Компьютеры" );
Если Не ПустоеЗначение( Компьютеры1 ) Тогда
    Для Х = 0 По Компьютеры1.КоличествоЭлементов() - 1 Цикл
        Если Не ПустоеЗначение( Компьютеры1.Элемент( Х ) ) Тогда
            Если Не ПустоеЗначение( Компьютеры1.Элемент( Х ).РабочиеМеста() ) Тогда
                РабочееМесто1 = Компьютеры1.Элемент(
                    Х ).РабочиеМеста().ПолучитьПоТипуСетевогоМеста( "Ядро" );
                Если Не ПустоеЗначение( РабочееМесто1 ) Тогда
                    Если РабочееМесто1.Локальное() == ИСТИНА Тогда // Требуемая проверка
                        Двери1 = РабочееМесто1.Двери();
                        Если Не ПустоеЗначение( Двери1 ) Тогда
                            Для У = 0 По Двери1.КоличествоЭлементов() - 1 Цикл
                                Если Не ПустоеЗначение( Двери1.Элемент( У ) ) Тогда
                                    Сообщить( Двери1.Элемент( У ).Название(), 1 );
                                    Считыватель1 = Двери1.Элемент( У ).СчитывательНаВход();
                                    Если Не ПустоеЗначение( Считыватель1 ) Тогда
                                        Сообщить( " " + Считыватель1.Название(), 1 );
                                        Считыватель1.ЗаблокироватьДоступ();
                                    КонецЕсли;
                                    Считыватель1 = Двери1.Элемент( У ).СчитывательНаВыход();
                                    Если Не ПустоеЗначение( Считыватель1 ) Тогда
                                        Сообщить( " " + Считыватель1.Название(), 1 );
                                        Считыватель1.ЗаблокироватьДоступ();
                                    КонецЕсли;
                                КонецЕсли;
                            КонецЦикла;
                        КонецЕсли;
                    КонецЕсли;
                КонецЕсли;
            КонецЕсли;
        КонецЕсли;
    КонецЕсли;
КонецЕсли;
КонецСценария

```

КонецЦикла;
КонецЕсли;
КонецСценария

Пример 4.

Сценарий
Переменные
Перем Компьютер1;
Перем Компьютеры1;
Перем РабочиеМеста1;
Перем Монитор1;
НачалоСценария

Попытка
Компьютеры1 = СоздатьОбъект("Компьютеры");
Если Не ПустоеЗначение(Компьютеры1) Тогда
Компьютер1 = Компьютеры1.ПолучитьПоИд(1);
Если Не ПустоеЗначение(Компьютер1) Тогда
РабочиеМеста1 = Компьютер1.РабочиеМеста();
Если Не ПустоеЗначение(РабочиеМеста1) Тогда
Монитор1 = РабочиеМеста1.ПолучитьПоТипуСетевогоМеста("Ядро");
Если Не ПустоеЗначение(Монитор1) Тогда
Монитор1.ВыслатьПисьмо("smtp.mail.ru","Orion","ORION","orion@mail.ru","Письмо
Орион","Привет мир!","mymail@mail.ru",465,1);
КонецЕсли;
КонецЕсли;
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;
КонецСценария

от

Пример 5.

Сценарий
Переменные
Перем Пароль1;
Перем Число1;
Перем Элемент1;
Перем Сотрудник1;
Перем Раздел1;
Перем КодСобытия1;
Перем Событие1;
Перем РабочееМесто1;
Перем РабочиеМеста1;
Перем Компьютер1;
Перем Компьютеры1;
Перем ОдинПрибор1;
НачалоСценария

Попытка
Компьютеры1 = СоздатьОбъект("Компьютеры");
Компьютер1 = Компьютеры1.ПолучитьПоИд(1);
РабочиеМеста1 = Компьютер1.РабочиеМеста();
РабочееМесто1 = РабочиеМеста1.ПолучитьПоТипуСетевогоМеста("Ядро");
Событие1 = ПолучитьСобытие();
Если Не ПустоеЗначение(Событие1) Тогда
КодСобытия1 = Событие1.Событие().Ид();
Элемент1 = Событие1.Элемент();
Если не ПустоеЗначение(Событие1.Пароль()) Тогда
Сотрудник1 = Событие1.Пароль().Сотрудник(); КонецЕсли;

```

Раздел1 = Событие1.Объект();
Пароль1 = Событие1.Пароль();
Если Не пустоеЗначение( РабочееМесто1.Приборы() ) Тогда
ОдинПрибор1 = РабочееМесто1.Приборы().ПолучитьПоИД( 29 );
Если Не пустоеЗначение( ОдинПрибор1 ) Тогда
Число1 = ТекущееВремяДатаЧисло();
ОдинПрибор1.ТранслироватьСобытие( КодСобытия1,Элемент1,Раздел1, Сотрудник1,Число1,
Пароль1);
КонецЕсли;
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение, шаг Отправка сообщения через УО-4с ",0);
КонецПопытки;
КонецСценария

```

Пример 6

Сценарий

Переменные

Перем РабочееМесто3;

Перем РабочиеМеста3;

Перем Компьютер3;

Перем Компьютеры3;

Перем Шлейф3;

Перем РабочееМесто2;

Перем РабочиеМеста2;

Перем Компьютер2;

Перем Компьютеры2;

Перем Шлейф2;

Перем Ядро1;

НачалоСценария

Ядро1 = СоздатьОбъект("Ядро");

Если Не ПустоеЗначение(Ядро1) Тогда

Ядро1.ОграничитьВыполнение(1);

КонецЕсли;

Попытка

Компьютеры2 = СоздатьОбъект("Компьютеры");

Компьютер2 = Компьютеры2.ПолучитьПоИД(1);

РабочиеМеста2 = Компьютер2.РабочиеМеста();

РабочееМесто2 = РабочиеМеста2.ПолучитьПоТипуСетевогоМеста("Ядро");

Если Не пустоеЗначение(РабочееМесто2.Шлейфы()) Тогда

Шлейф2 = РабочееМесто2.Шлейфы().ПолучитьПоИД(165);

Если Не пустоеЗначение(Шлейф2) Тогда

Шлейф2.Взять();

КонецЕсли;

КонецЕсли;

Исключение Сообщить("Произошло исключение, шаг Взять на охрану ШС сообщение ",0);

КонецПопытки;

Попытка

Компьютеры3 = СоздатьОбъект("Компьютеры");

Компьютер3 = Компьютеры3.ПолучитьПоИД(1);

РабочиеМеста3 = Компьютер3.РабочиеМеста();

РабочееМесто3 = РабочиеМеста3.ПолучитьПоТипуСетевогоМеста("Ядро");

Если Не пустоеЗначение(РабочееМесто3.Шлейфы()) Тогда

Шлейф3 = РабочееМесто3.Шлейфы().ПолучитьПоИД(160);

Если Не пустоеЗначение(Шлейф3) Тогда

Шлейф3.Взять();

КонецЕсли;

КонецЕсли;

Исключение Сообщить("Произошло исключение, шаг Взять на охрану ШС сообщение ",0);
КонецПопытки;

Пример 7

Сценарий

Переменные

Перем Компьютеры1;

Перем Компьютер1;

Перем Пароли1;

Перем ЧислоПароль1;

Перем ЗоныДоступа1;

Перем ЗонаДоступа1;

Перем Пароль1;

НачалоСценария

Компьютеры1 = СоздатьОбъект("Компьютеры");

Компьютер1 = Компьютеры1.ПолучитьПоИД(7);

Пароли1 = СоздатьОбъект("Пароли");

ЗоныДоступа1 = СоздатьОбъект("ЗоныДоступа");

ЗонаДоступа1 = ЗоныДоступа1.ПолучитьПоИндексу(9); //определяем зону доступа 0 - "Внешний Мир"

ЧислоПароль1 = ЗонаДоступа1.ПарольПоследнего(); // определяем пароль последнего прошедшего в зону "Внешний Мир"

Пароль1 = Пароли1.ПолучитьпоИД(ЧислоПароль1);

Если не ПустоеЗначение(Пароль1) тогда

Сообщить(Пароль1.Сотрудник().ФИО(),1);

Пароль1.Заблокировать(); // Блокируем пароль последнего . тут какая-то ошибка

Задержка(3600); // ставим паузу

Пароль1.Разблокировать(); // разблокируем пароль

иначе

Сообщить("нет объекта", 1);

КонецЕсли;

КонецСценария

Пример 8

Сценарий

Переменные

Перем ПочтовыйЯщик5;

Перем Почта5;

Перем РабочееМесто5;

Перем РабочиеМеста5;

Перем Компьютер5;

Перем Компьютеры5;

Перем ПочтовыйЯщик4;

Перем Почта4;

Перем РабочееМесто4;

Перем РабочиеМеста4;

Перем Компьютер4;

Перем Компьютеры4;

Перем ПочтовыйЯщик3;

Перем Почта3;

Перем РабочееМесто3;

Перем РабочиеМеста3;

Перем Компьютер3;
Перем Компьютеры3;
Перем ПочтовыйЯщик2;
Перем Почта2;
Перем РабочееМесто2;
Перем РабочиеМеста2;
Перем Компьютер2;
Перем Компьютеры2;
Перем Логическое1;
Перем Почта1;
Перем РабочееМесто1;
Перем РабочиеМеста1;
Перем Компьютер1;
Перем Компьютеры1;
НачалоСценария

Попытка

Компьютеры1 = СоздатьОбъект("Компьютеры");
Компьютер1 = Компьютеры1.ПолучитьПоИд(1);
РабочиеМеста1 = Компьютер1.РабочиеМеста();
РабочееМесто1 = РабочиеМеста1.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто1) Тогда
Почта1 = РабочееМесто1.Почта();
Логическое1 = Логическое(0);
Почта1.ДобавитьПочту("smtp.mail.ru","aaa", "orion", 465, Логическое1, 0);
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;

Попытка

Компьютеры2 = СоздатьОбъект("Компьютеры");
Компьютер2 = Компьютеры2.ПолучитьПоИд(1);
РабочиеМеста2 = Компьютер2.РабочиеМеста();
РабочееМесто2 = РабочиеМеста2.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто2) Тогда
Почта2 = РабочееМесто2.Почта();
ПочтовыйЯщик2 = Почта2.ПолучитьПочту("smtp.mail.ru");
Если не ПустоеЗначение(ПочтовыйЯщик2) Тогда
ПочтовыйЯщик2.РазрешитьПочту("111@bolid.ru");
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;

Попытка

Компьютеры3 = СоздатьОбъект("Компьютеры");
Компьютер3 = Компьютеры3.ПолучитьПоИд(1);
РабочиеМеста3 = Компьютер3.РабочиеМеста();
РабочееМесто3 = РабочиеМеста3.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто3) Тогда
Почта3 = РабочееМесто3.Почта();
ПочтовыйЯщик3 = Почта3.ПолучитьПочту("smtp.mail.ru");
Если не ПустоеЗначение(ПочтовыйЯщик3) Тогда
ПочтовыйЯщик3.РазрешитьПочту("222@bolid.ru");
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;

Попытка

Компьютеры4 = СоздатьОбъект("Компьютеры");
Компьютер4 = Компьютеры4.ПолучитьПоИд(1);
РабочиеМеста4 = Компьютер4.РабочиеМеста();
РабочееМесто4 = РабочиеМеста4.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто4) Тогда
Почта4 = РабочееМесто4.Почта();

```
ПочтовыйЯщик4 = Почта4.ПолучитьПочту("smtp.mail.ru");
Если не ПустоеЗначение(ПочтовыйЯщик4) Тогда
ПочтовыйЯщик4.РазрешитьПочту("333@gmail.com");
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;
```

```
Попытка
Компьютеры5 = СоздатьОбъект( "Компьютеры" );
Компьютер5 = Компьютеры5.ПолучитьПоИд(1);
РабочиеМеста5 = Компьютер5.РабочиеМеста();
РабочееМесто5 = РабочиеМеста5.ПолучитьПоТипуСетевогоМеста("Ядро");
Если не ПустоеЗначение(РабочееМесто5) Тогда
Почта5 = РабочееМесто5.Почта();
ПочтовыйЯщик5 = Почта5.ПолучитьПочту("smtp.mail.ru ");
Если не ПустоеЗначение(ПочтовыйЯщик5) Тогда
ПочтовыйЯщик5.ПринятьПочту();
КонецЕсли;
КонецЕсли;
Исключение Сообщить("Произошло исключение",0); КонецПопытки;
КонецСценария
```

Пример 9

```
СЦЕНАРИЙ
ПЕРЕМЕННЫЕ
ПЕРЕМ label1;
ПЕРЕМ label2;
НАЧАЛОСЦЕНАРИЯ
label2 = "c:\soundorion.wav";
label1 = СоздатьОбъект( "ЯДРО" );
label1.КонвертироватьСообщениеВФайл("Это Сообщение было сконвертировано IN label2 с
помощью звукового помощника Орион. Ура!", Файл);
label1.ВОСПРОИЗВЕСТИФАЙЛ( label2 );
КОНЕЦСЦЕНАРИЯ
```

Пример 10.

```
СЦЕНАРИЙ
ПЕРЕМЕННЫЕ
ПЕРЕМ label1;
НАЧАЛОСЦЕНАРИЯ
label1 = СоздатьОбъект( "ЯДРО" );
label1.ВоспроизвестиОповещение("Привет!", 15, Истина, 3);
КОНЕЦСЦЕНАРИЯ
```

Пример 11

```
СЦЕНАРИЙ
ПЕРЕМЕННЫЕ
ПЕРЕМ label1;
ПЕРЕМ label2;
НАЧАЛОСЦЕНАРИЯ
label1 = СоздатьОбъект( "ОБОЛОЧКА" );
label1.ЗаписатьВЖурналСобытий("Привет 1!");
label2 = СоздатьОбъект( "МОНИТОР" );
label2.ЗАПУСТИТЬХРАНИТЕЛЬЭКРАНА();
label1.ЗаписатьВЖурналСобытий("Привет 2!");
КОНЕЦСЦЕНАРИЯ
```