:АЛГОРИТМЫ ПЕРЕМЕННЫЕ ФУНКЦИИ

КОНСТРУКТОР БИЗНЕС-ПРОЦЕССОВ РЕДАКЦИЯ 2.0

ДОБРО ПОЖАЛОВАТЬ В МИР БИЗНЕС-ПРОЦЕССОВ.

Руководство пользователя.

Мы рады, что вы решили приобрести систему Конструктор Бизнес-Процессов.

Чем лучше вы будете знакомы с её возможностями, тем увереннее и эффективнее сможете использовать её в своей работе. Поэтому мы просим:

Прочтите руководство пользователя до того, как начнете использовать систему Конструктор Бизнес-Процессов. Также воспользуйтесь интегрированной в программу справкой. Здесь содержатся важные сведения по управлению функционалом, которые помогут вам максимально эффективно использовать все технические преимущества системы. Кроме того, в документации вы найдете информацию, которая способствует надежной и безопасной работе программы, а также поможет сохранить её производительность и стабильность на высоком уровне.

Печатная версия руководства пользователя соответствует актуальному состоянию возможностей программы. После выпуска обновлений руководство пользователя также обновляется актуальной информацией.

Дополнительные материалы и инструкции вы можете найти в других документах, входящих в комплект поставки, а также на сайте: https://www.1bpm.ru.

Желаем вам эффективного использования системы КонструкторБизнесПроцессов.

Техническая поддержка

Для пользователей программного продукта доступна техническая поддержка.
По всем вопросам следует обращаться на адрес: mail@1bpm.ru.
Наша команда всегда рада вам помочь.

© 2025 Общество с ограниченной ответственностью «Новые Системы», Иркутск, Россия
Воспроизведение, полное или частичное, допускается только
с письменного разрешения ООО «Новые Системы», Иркутск.

Все исключительные права на программный продукт
«Новые Системы: Конструктор Бизнес-Процессов. Редакция 2.0»
и документацию принадлежат ООО «Новые Системы», 2015–2025 г.
РосПатент: №2022667718 от 23.09.22
Единый реестр: №15564 от 18.11.22
Сайт: https://www.1bpm.ru, e-mail: mail@1bpm.ru

Актуально для версий:
Основная поставка: 2.0.6.13
Расширение «Сигнал»: 2.0.2.1

Наименование книги: «Использование алгоритмов, переменных и функций в программе КонструкторБизнесПроцессов Редакция 2.0. Руководство пользователя»
Дата первого выхода: 27.10.2022 г.
Дата выхода текущей версии: 10.06.2025 г.

Содержание

Введение ... 6
К этому руководству ... 6
Термины ... 6
Дополнительные источники информации ... 6
Всегда на связи ... 6
Таблица использования алгоритмов ... 7
Переменные ... 8
Процедуры и функции ... 17
ДанныеШага ... 17
НайтиЗадачу ... 18
СостояниеЗадачи ... 18
ДанныеПроцесса ... 19
НовоеСостояние ... 19
_СохранитьЗначение ... 20
_ПолучитьЗначение ... 20
ВыполнитьЗадачу ... 21
ЗапуститьНовыйПроцесс ... 21
РассчитатьПлановуюДатуВыполнения ... 22
Процедуры и функции расширения «Сигнал» ... 24
нс_Телеграм.ПолучитьДанныеБуфера ... 24
нс_Телеграм.СохранитьВложения ... 24
Описание алгоритмов процессов и задач ... 25
Алгоритм при запуске нового процесса ... 25
Основной обработчик процесса ... 25
Алгоритм при завершении процесса ... 25
Алгоритм расчета срока выполнения ... 26
Алгоритм при запуске новой задачи ... 26
Алгоритм автозавершения задачи ... 26
Алгоритм заполнения таблицы данных ... 27
Вызов алгоритмов по имени ... 28
Динамические команды на форме задачи ... 29
Встроенный редактор программного кода ... 31
Приложения ... 33
Пример программной настройки меню выбора для чат-ботов Telegram ... 33
Дополнительные возможности программной настройки чат-ботов Telegram ... 36
Обработка неизвестного запроса ... 36
Сохранение вложений ... 37
Переадресация обработки входящего сообщения ... 38

Введение

К этому руководству

В настоящем Руководстве приведено описание алгоритмов, переменных и функций, используемых в программе КонструкторБизнесПроцессов© (далее Система).

Руководство является дополнением к основному Руководству пользователя по программе и Руководству пользователя по расширению «Сигнал», входящему в комплект основной поставки.

Применение программного кода для настройки сценариев не обязательно, но именно он позволяет использовать Систему максимально эффективно.

Термины

- Алгоритм – элемент справочника *Алгоритмы*, содержащий исполняемый программный код и дополнительные настройки.
- Программный код – исполняемый программный код, не хранимый в справочнике *Алгоритмы*.

Дополнительные источники информации

Дополнительные материалы (статьи, примеры, ответы на вопросы, видео) доступны на официальном сайте.

Сайт: https://www.1bpm.ru.

Всегда на связи

Адрес электронной почты для технической поддержки, общих вопросов и предложений: mail@1bpm.ru.

УВЕДОМЛЕНИЕ

Автор оставляет за собой право на опечатки, ошибки и технические изменения.

Копирование и распространение текста (всего или отдельных его частей) в любом виде и для любых целей без разрешения автора запрещено.

Таблица использования алгоритмов

В таблице перечислены места использования алгоритмов (А) и программного кода (П).

Объект	Использование		А	П
Сценарии	Автозапуск по произвольному условию		•	•
	Динамический отбор объекта процесса			•
	Динамические отборы в таблице объектов процесса			•
	Параметры шаблона наименования процесса			•
	Алгоритм, выполняемый при создании процесса		•	
	Параметры описаний шагов сценария		•	•
	Отложенный запуск шагов сценария по условию		•	•
	Действие	Параметры шаблона наименования	•	•
		Настройка динамических кнопок в задачах	•	•
		Алгоритм, выполняемый при запуске	•	
		Условие автоматического и ручного завершения	•	
		Динамический отбор объекта		•
		Динамические отборы в таблице объектов		•
		Настройка адресации	•	•
	Обработка	Тело обработки	•	•
	Условие	Автоматическая проверка условий	•	•
	Выбор	Автоматический выбор вариантов	•	•
Причины	Остановки	Условия автоматических остановок	•	
	Отмены	Условия автоматической отмены	•	
«Сигнал»	Настройка оповещений	Определение получателей оповещений	•	
		Параметры шаблона текста оповещений		•
		Определение контекста для системы взаимодействия	•	
	Чат-боты Telegram	Дополнительные алгоритмы авторизации	•	
		Шаблон запроса для получения дополнительных вложений		•
	Ключевые слова	Алгоритмы обработки входящих сообщений/запросов	•	

Переменные

Главные список

- Для ссылочных значений доступно получение реквизитов через точку. Пример: _Процесс.Сценарий.

Имя	Тип	Описание
:Переменные общего назначения		
_Процесс	бип_Процесс	Текущий процесс.
_ПроцессОбъект	бип_ПроцессОбъект	Текущий процесс. Доступен только в Алгоритмах, выполняемых при запуске новых процессов.
_ТаблицаДанных	ТаблицаЗначений	Таблица данных текущего процесса. Заполняется в Алгоритм заполнения таблицы данных и Алгоритмах, выполняемых при запуске новых процессов, если в сценарии настроено использование таблицы данных. - Доступ к таблице при автоматическом запуске через переменную _ТаблицаДанных. - Доступ к таблице в алгоритме заполнения при ручном запуске процесса через переменную _ПроцессОбъект.ТаблицаДанных. - В прочих случаях, доступ к таблице данных возможен через _Процесс.ТаблицаДанных.

Имя	Тип	Описание
_Процесс.Объект	СправочникСсылка ДокументСсылка	Основной объект процесса. Если процесс запускается *по событию*, в реквизите хранится ссылка на элемент справочника или документ, по которому произошло событие.
_Процесс.Основание	бип_Процесс	Родительский процесс текущего процесса.
_Задача	бип_Задача	Предшествующая текущему шагу задача. Текущая задача, если выполняется Алгоритм при запуске задачи или Алгоритм проверки автоматического завершения задачи по условию.
_ВыполненаВручную	Булево	Признак интерактивного выполнения задачи (нажатием кнопки *Выполнена*). Используется только в Алгоритм проверки автоматического завершения задачи по условию.
_ИмяШага	Строка	Имя текущего шага как оно задано в настройках сценария. Например, *&Действие1*. Используется в *Алгоритмах настроек адресации*.
_ПредставлениеШага	Строка	Представление текущего шага как оно задано в настройках сценария. Используется в *Алгоритмах настроек адресации*.
_ПараметрN (N = 1,2,3,4,5)	Произвольный	Фиксированное, константное значение, указанное в настройках сценария.

Имя	Тип	Описание
_ПараметрNИмя (N = 1,2,3,4,5)	Строка	Имя фиксированного, константного значения, указанного в настройках сценария.
_Исполнитель	Строка Пользователи	Исполнитель текущей задачи. Доступен только в Алгоритмах, выполняемых при запуске новых задач и при использовании алгоритмов в расширении *Сигнал*: ▷ Для программного определения получателей оповещений. ▷ При выполнении алгоритмов в чат-ботах. В этом случае в переменной содержится значение текущего идентифицированного пользователя *Telegram*. В прочих случаях для получения исполнителя задачи следует использовать функцию НайтиЗадачу().
_Процесс.Автор _Задача.Автор _Задача.Исполнитель _Задача.Наблюдатель	Строка Пользователи Роли НаборыИсполнителей	Автор процесса. Автор задачи. Исполнитель задачи. Наблюдатель задачи.
_ДанныеДляОбработки	Соответствие	Переменная используется при обработке данных, полученных от чат-ботов расширения *Сигнал*. ▷ Содержит *историю* текущего взаимодействия (контекст «беседы») – результаты выбора из меню, введенный комментарий, прочие

Имя	Тип	Описание
		сведения, в т.ч. добавляемые программно через алгоритмы ключевых слов. ▷ Очищается при обработке ключевого слова с видом *Текстовый ответ* (если не включена настройка ключевого слова *Сохранять контекст*). ▷ Очищается при выполнении алгоритмов ключевых слов вида *Выполнение алгоритма* (если значение переменной _Результат не является массивом или структурой). ▷ Очищается при отказе от подтверждения выбора для ключевых слов вида *Выбор*. Также, см.: ▷ Руководство пользователя «Сигнал» (раздел *Ключевые слова*). ▷ Пример программной настройки меню выбора для чат-ботов Telegram. ▷ Дополнительные возможности программной настройки чат-ботов Telegram.
_Результат	Булево Произвольный	Переменная, в которую помещается результат вычисления какого-либо значения. Используется при проверке условий, программном определении исполнителей задач и т.п. ▷ При использовании алгоритма как процедуры или функции (см. Вызов алгоритмов по имени), переменная доступна при вызове каждого

Имя	Тип	Описание
		алгоритма. Таким образом, переменную можно использовать для передачи значения при вызове нескольких алгоритмов. ▷ При программном определении исполнителей задач в переменную _Результат передаётся значение *Исполнителя*, *Роли исполнителя* или *Массив*, содержащий исполнителей и/или роли. ▷ При этом для групповых задач, если *Массив* содержит *Роль исполнителя*, то исполнителем задачи устанавливается *Роль*, а не пользователи, обладающие данной ролью. ▷ Если в переменную передана *Структура* вида Новый Структура("ОбъектыПоИсполнителям, Исполнители", Истина, Массив), то каждая групповая задача будет привязана к объекту процесса из табличной части *Объекты*, если количество исполнителей совпадает с количеством объектов в процессе. ▷ Пример: *Массив* содержит значения *Иванов*, *Петров*, *Логисты*. При обработке *группового* шага сценария будет создано 3 задачи: задача для *Иванова*, задача для *Петрова* и задача для *Логистов*. Любой из логистов сможет взять задачу для исполнения.

Имя	Тип	Описание
		▷ При использовании *Массива* для *не групповой* задачи, в массиве учитываются только исполнители, роли в массиве игнорируются. ▷ В расширении *Сигнал* используется для передачи результатов выполнения алгоритмов для ключевых слов вида *Выполнение алгоритма*. Также, см.: ▷ Руководство пользователя «Сигнал» (раздел *Ключевые слова*). ▷ Пример программной настройки меню выбора для чат-ботов Telegram.
:Переменные проверки условий		
_Результат	Булево	Условие проверяется по значению переменной. Условие выполнено, если _Результат = Истина.
_РасшифровкаРезультата	Строка	В переменной может быть указана дополнительная информация о результатах проверки условия или выбора вариантов. Значение переменной выводится: ▷ В сообщении пользователю при выполнении задачи с включенной настройкой Автоматически завершать задачу по условию. ▷ На карте процесса, как дополнительная информация для шагов вида *Условие* и *ВыборВарианта*.

Имя	Тип	Описание
:Переменные автозапуска процессов по событиям		
_Источник	СправочникСсылка ДокументСсылка	Источник возникновения события, которое привело к запуску текущего процесса.
_ИсточникДо	Структура	Данные (стандартные реквизиты, реквизиты и табличные части) источника возникновения события, которое привело к запуску текущего процесса, до его текущего изменения (записи). Используется для запуска процессов при изменении реквизитов источника. Пример: _Результат = _ИсточникДо.Сумма <> _Источник.Сумма.
_ЭтоНовый	Булево	Признак первой записи источника события (Признак создания).
_ПользовательСобытия	Строка Пользователи	Пользователь, под которым произошло событие.

Имя	Тип	Описание
:Переменные автозапуска процессов по условию		
_ВыходныеДанные	Массив	Массив, в который могут быть добавлены произвольные данные. Используется для передачи данных в создаваемые на основании сценария процессы.
_РазделятьПоПроцессам	Булево	Признак создания нескольких процессов. Каждому элементу массива _ВыходныеДанные соответствует один создаваемый процесс. Если _РазделятьПоПроцессам = Ложь, все объекты из массива _ВыходныеДанные будут записаны в табличную часть *Объекты* нового процесса.
Параметры.Объект	Произвольный	Значение, которое будет указано в качестве объекта нового процесса.
Параметры.Сценарий	бип_Сценарии	Текущий сценарий.
:Переменные расширения *Сигнал* и параметров в текстах оповещений		
_ПолучательОповещения	СправочникСсылка	Используется в алгоритмах оповещений и при расчете программных параметров в текстах оповещений.
_ТекстОповещения	Строка	Используется в алгоритмах оповещений. Содержит текст оповещения.
_ТекстОповещенияHTML	Строка	Используется в алгоритмах оповещений. Содержит текст оповещения в формате HTML.

Дополнительные переменные

- Для шагов вида *Условие* и *ВыборВарианта*: для получения данных о пройденных шагах сценария можно обращаться к ним по имени в виде &Действие1, &Условие2 и т.п.
- Имена шагов сценария можно узнать, дважды щёлкнув по любой из линий связи, относящихся к шагу, или наведя курсор на элемент графической схемы.

Имя	Тип	Описание
:Получение дополнительных сведений о шагах сценария		
.ДатаСоздания	Дата	Дата создания шага.
.ДатаВыполнения	Дата	Дата выполнения шага.
.Просрочено	Булево	Признак просроченной задачи.
.Результат	Строка	Результат проверки *Условия* (Да/Нет). Результат выбора на шаге *ВыборВарианта*.
.КоличествоПовторов	Число	Количество повторов текущего шага. Первое выполнение шага повтором не является.

Процедуры и функции

В разделе перечислены процедуры и функции, которые можно использовать для работы с объектами Системы.

Структура:

- Имя процедуры / функции.
- Описание.
- Параметры.
- (для функций) Возвращаемое значение.

В скобках после каждого параметра и результата выполнения указан тип значения.

ДанныеШага()

Возвращает структуру данных шага процесса по его имени или порядковому номеру на карте процесса.

Если по имени найдено несколько шагов, то функция возвращает массив структур.

Параметры:

- Процесс (бип_Процесс) – Процесс, к которому относится шаг.
- ИмяНомер (Строка, Число) – Имя шага сценария или номер шага на карте процесса. Например, «&Действие1» или 3.

Возвращаемое значение:

- Структура – состоит из:
  - Период (Дата) – Дата шага.
  - Процесс (бип_Процесс) – Процесс.
  - НомерШага (Число) – Номер шага.
  - Имя (Строка) – Имя шага.
  - УсловиеВыбор (Строка) – Результат обработки выбора или условия.
  - Действие (бип_Задача, бип_Процесс) – Задача или вложенный процесс.
  - ОтложеннаяДата (Дата) – Дата отложенного запуска шага.
  - РасшифровкаРезультата (Строка) – Подробное описание результата проверки условия.
- Массив.

НайтиЗадачу()

Возвращает структуру, содержащую ссылку на задачу или вложенный процесс по указанному имени, относящиеся к указанному процессу.

Дополнительно, в структуре содержится информация об авторе и исполнителе задачи.

Если по имени найдено несколько задач или вложенных процессов, то функция возвращает массив структур.

Параметры:

- Процесс (бип_Процесс) – Процесс, к которому относится задача.
- Имя (Строка) – имя шага сценария в виде:
  - «&ДействиеN».
  - шаблона для поиска нескольких объектов.

  По шаблону «&ВложенныйПроцесс%» будут найдены все дочерние процессы).

Возвращаемое значение:

- Структура – состоит из:
  - Задача (бип_Задача, бип_Процесс) – Найденная задача или вложенный процесс.

    Если задача или процесс не найдены, возвращается пустое значение бип_Задача.ПустаяСсылка.
  - Автор (Пользователи) – Автор задачи.
  - Исполнитель (Пользователи) – Текущий исполнитель задачи.
- Массив.

СостояниеЗадачи()

Возвращает состояние задачи на указанную дату.

Параметры:

- Задача (бип_Задача) – Задача.
- НаДату (Дата, Неопределено) – Дата получения состояния. Если не указана, будет получено текущее состояние.

Возвращаемое значение:

- Структура – состоит из:
  - Период (Дата) – Дата установки текущего состояния.
  - Состояние (бип_СостоянияЗадач) – Текущее состояние задачи.

- ПричинаОстановкиОтмены (бип_ПричиныОстановок, бип_ПричиныОтмены) – Причина остановки / отмены задачи.
- Комментарий (Строка) – Дополнительный комментарий при остановке / отмене задачи.
- НомерШага (Число) – Порядковый номер шага задачи в истории процесса.

ДанныеПроцесса()

Возвращает структуру, содержащую данные указанного процесса.

Параметры:

- Процесс (бип_Процесс) – Процесс, к которому относится задача.

Возвращаемое значение:

- Структура – состоит из:
  - Ссылка (бип_Процесс) – Ссылка на процесс.

    Если процесс не найден (передана некорректная ссылка), то возвращается бип_Процесс.ПустаяСсылка.
  - Автор (Пользователи) – Автор процесса.
  - Период (Дата) – Дата установки текущего состояния.
  - Состояние (бип_СостоянияПроцессов) – Текущее состояние процесса.
  - ПричинаОстановкиОтмены (бип_ПричиныОстановок, бип_ПричиныОтмены) – Причина остановки / отмены процесса.
  - Комментарий (Строка) – Дополнительный комментарий при остановке / отмене процесса.

НовоеСостояние()

Устанавливает новое состояние для процесса или задачи.

Параметры:

- Объект (бип_Процесс, бип_Задача) – Ссылка на процесс или задачу.
- Состояние (бип_СостоянияПроцессов, бип_СостоянияЗадач) – Новое состояние.
- Период (Неопределено, Дата) – Дата установки нового состояния.
- ПричинаОстановкиОтмены (бип_ПричиныОстановок,

бип_ПричиныОтмены) – Причина остановки / отмены процесса / задачи.

- Комментарий (Неопределено, Строка) – Дополнительный комментарий при остановке / отмене процесса / задачи.
- ТекстОшибки (Неопределено, Строка) – Переменная, в которую возвращается текст ошибки. Если не указана, ошибка записывается в журнал регистрации.

Возвращаемое значение:

- Булево:
  - Истина – Новое состояние установлено.
  - Ложь – Ошибка при установки нового состояния.

_СохранитьЗначение()

Сохраняет переданное значение по указанному имени. Значение используется только в рамках указанного процесса.

Параметры:

- Имя (Строка) – Имя, под которым сохраняется значение.
- Значение (Произвольный) – Сохраняемое значение.

  Если не заполнено, сохраненное ранее по указанному имени значение удаляется.
- Процесс (бип_Процесс, Неопределено) – Процесс, для которого сохраняется значение.

  Если не указан, значение сохраняется для текущего процесса.

_ПолучитьЗначение()

Возвращает сохраненное значение по указанному имени для указанного процесса. Если процесс не указан, возвращается значение текущего процесса.

Параметры:

- Имя (Строка) – имя получаемого значения.
- Процесс (бип_Процесс, Неопределено) – Процесс, для которого получается значение.

  Если не указан, то для текущего процесса.

Возвращаемое значение:

- Произвольное – Полученное значение.

- Неопределено – Если значение по имени не найдено.

бип_НаСервере.ВыполнитьЗадачу()

Используется для программного выполнения задачи. Например, при встраивании выполнения задач в объекты основной конфигурации.

Параметры:

- Задача (бип_Задача, Строка) – Ссылка на задачу или имя шага сценария в виде «&ДействиеN».
- Процесс (бип_Процесс, Неопределено) – Используется, если в параметр Задача передано имя шага сценария.
- ЗаполнитьРеквизитыПоУмолчанию (Булево) – Используется для автоматического заполнения реквизитов.
- Обязательные для заполнения (но незаполненные) реквизиты задачи будут заполнены:
  - Условие – «Нет».
  - Выбор варианта – последний вариант.
  - Вариант из списка – последний из списка вариантов.
- ПроверятьЗаполнениеРеквизитов (Булево) – Признак проверки реквизитов задачи.

  Если Ложь, задача будет завершена без проверок с признаком *Завершена автоматически*.

Возвращаемое значение:

- Структура – состоит из:
  - Выполнена (Булево) – Результат выполнения задачи:
    - Истина – Задача выполнена.
    - Ложь – Ошибка при выполнении задачи.
  - Расшифровка (Строка) – Описание ошибки выполнения задачи. Если задача выполнена, не заполняется.

бип_Автозапуск.ЗапуститьНовыйПроцесс()

Используется для программного создания и запуска нового процесса по указанному сценарию
и объекту / объектам.

Может использоваться в алгоритмах Системы и / или в прочих случаях, когда требуется

программное создание и запуск новых процессов, например в http-сервисах.

Параметры:

- Параметры (Структура) – состоит из:
  - Сценарий (бип_Сценарий) – Сценарий, по которому запускается процесс.
  - Дата (Дата) – Дата и время нового процесса.
  - Событие (бип_ТипыСобытий) – Событие запуска.
  - Комментарий (Строка) – Дополнительный комментарий к процессу.
- Объект (Неопределено, Число, Строка, Дата, Булево, ЛюбаяСсылка) – Объект, который записывается в процесс.

  Если не указан, запущенный процесс не будет связан ни с каким объектом системы.
- Объекты (Массив, Неопределено) – Массив объектов, записываемых в процесс в табличную часть *Объекты*.

Возвращаемое значение:

- Структура – состоит из:
  - Процесс (бип_Процесс) – Ссылка на новый процесс.
  - ТекстОшибки (Строка) – Описание ошибки запуска нового процесса. Если процесс запущен, не заполняется.

бип_НаСервере.РассчитатьПлановуюДатуВыполнения()

Используется для расчета срока выполнения от указанной даты.

Если указан параметр *Календарь*, выходные дни и почасовое расписание берутся из него.

Если *Календарь* не указан, определяется по константе БСП *Основной календарь предприятия*.

Если не удается получить основной календарь или календарь не заполнен, то выходными считаются только суббота и воскресенье.

Если не заполнено расписание календаря (или календарь не определен), то начало и окончание рабочего дня берутся из модуля повторного использования.

Также, см. Алгоритм, выполняемый после расчета срока выполнения процесса или задачи.

Параметры:

- Дата (Дата) – Начальная дата.
- Параметры (Структура) – состоит из:
  - Срок (Число) – Срок в секундах.
  - СрокДата (Дата) – Срок в виде конкретной даты.
  - СрокКонецДня (Булево) – Признак установки срока в конце рабочего дня.
  - НеУчитыватьВыходные (Булево) – Признак учета выходных при расчете срока.
  - ТолькоРабочиеДни (Булево) – Признак расчета срока только по рабочим дням.
- Календарь (СправочникСсылка.Календари, Неопределено) – Календарь.

Возвращаемое значение:

- Дата – Рассчитанная плановая дата.

Процедуры и функции расширения «Сигнал»

нс_Телеграм.ПолучитьДанныеБуфера()

Возвращает строковое представление переданного соответствия.

Параметры:

- ДанныеДляОбработки (Соответствие) – История текущего взаимодействия.

Возвращаемое значение:

- Строка

нс_Телеграм.СохранитьВложения()

Сохраняет вложения из полученного сообщения пользователя.

Результаты сохранения записываются в переданный параметр ДанныеДляОбработки, с ключами СохранениеФайла, ПутьКФайлу и error.

См. Сохранение вложений.

Параметры:

- ДанныеДляОбработки (Соответствие) – История текущего взаимодействия.
- Каталог (Строка) – Каталог сохранения.
- РаздельноеХранение (Булево) – Признак раздельного сохранения вложений по папкам в виде *Каталог\КлючПолучателя\Дата* (в формате *ДФ=yyyyMMddHHmmss*).

  Значение по умолчанию: Истина.

Описание алгоритмов процессов и задач

Алгоритм, выполняемый при запуске нового процесса

Путь

Форма сценария – Закладка *Дополнительно* – поле *При создании процесса выполнять алгоритм*.

Описание

Алгоритм выполняется после начала транзакции записи нового процесса, но до начала его записи в процедуре ПередЗаписью() модуля документа.

Обращение к объекту текущего процесса возможно через переменную _ПроцессОбъект (бип_ПроцессОбъект).

Алгоритм – Основной обработчик процесса

Путь

Форма сценария – Закладка *Дополнительно* – поле *Основной обработчик*.

Описание

Алгоритм «*постоянного действия*».

Выполняется для каждого процесса каждый раз при выполнении основного регламентного задания.

Алгоритм, выполняемый при завершении процесса

Путь

Форма сценария – Закладка *Дополнительно* – поле *При завершении процесса выполнять алгоритм*.

Описание

Алгоритм выполняется в транзакции записи статуса Завершен.

Алгоритм, выполняемый после расчета срока выполнения процесса или задачи

Путь

Форма сценария – Форма настройки срока выполнения процесса или действия – поле *После расчета срока выполнить алгоритм*.

Описание

Алгоритм выполняется после начала транзакции записи нового процесса или задачи, но до начала их записи в процедуре ПередЗаписью() модуля соответствующего документа.

Обращение к процессу возможно через переменную _ПроцессОбъект (бип_ПроцессОбъект).

Обращение к задаче возможно через переменную _Задача (бип_ЗадачаОбъект).

Алгоритм, выполняемый при запуске новой задачи

Путь

Форма сценария – Форма настройки запуска действия – поле *При запуске выполнять алгоритм*.

Описание

Алгоритм выполняется после записи новой задачи и после записи дополнительных данных в регистры сведений (данные об авторе, исполнителях, наблюдателях, статусе текущей задачи и технических данных по процессу).

Обращение к ссылке на задачу возможно через переменную _Задача (бип_Задача).

Алгоритм проверки завершения / автозавершения задачи

Путь

Форма сценария – Форма настройки действия – поле *Алгоритм проверки условия* при включенной настройке *Завершать задачу по условию*.

Описание

Алгоритм выполняется при записи задачи (при выполнении) и в регламентном задании подсистемы, проверяющем условия автозавершения задач.

Задача не сможет быть завершена, если переменная _Результат = Ложь.

Сообщение пользователю будет содержать одно из значений:

- значение переменной _РасшифровкаРезультата.
- комментарий к алгоритму.
- наименование алгоритма.

Обращение к задаче возможно через переменные:

- _Задача (бип_Задача) в регламентном задании.
- _Задача (бип_ЗадачаОбъект) при записи задачи.

Алгоритм заполнения таблицы данных

Путь

Форма сценария – Закладка *Дополнительно* – форма *Настройка таблицы данных* при включенной настройке *Использовать таблицу данных*.

Описание

Алгоритм выполняется только при ручном создании процесса. Доступ к таблице через переменную _ПроцессОбъект.ТаблицаДанных.

При автоматическом создании процесса для заполнения таблицы используйте Алгоритм, выполняемый при запуске нового процесса. Доступ к таблице данных через переменную _ТаблицаДанных.

Имена колонок таблицы указываются при настройке таблицы данных.

Вызов алгоритмов по имени

Общая информация

Система позволяет выполнять программный код алгоритмов, вызывая их по имени.

Это позволяет многократно использовать программный код в разных местах в комбинации с другими алгоритмами.

Имя алгоритма

Имя алгоритма указывается без пробелов.

Пример: *АлгоритмПроверки*.

Вызов

При вызове алгоритма следует добавлять знак #.

Пример: #АлгоритмПроверки

Параметры

При вызове алгоритма по имени могут быть переданы параметры.

Параметры указываются через запятую.

Пример: *#АлгоритмПроверки* (_Процесс, "Действие1", 1, Истина)

В качестве параметров могут быть переданы только значения и общие переменные Системы (_Процесс, _Источник и т.д.).

Значение переданных параметров в коде алгоритма доступны через переменные Параметр1, Параметр2, Параметр3 и т.д.

Динамические команды на форме задачи

Общая информация

- Динамические команды на форме задачи могут использоваться для разных целей.

  Например, для открытия формы отчёта по связанному с задачей объекту.
- Программный код и код алгоритмов динамических команд, настроенных в сценарии для задач, выполняется *&НаКлиенте*.
- Переменные подсистемы (_Процесс, _Задача и т.д.) не инициализируются и не доступны, т.к. все необходимые значения можно получить из реквизитов формы.
- Программный код следует использоваться при вычислении простых выражений.

  Для многострочного кода следует использовать алгоритмы.
- Для получения реквизитов объектов в конфигурациях, разработанных с использованием БСП, следует использовать типовые функции общего модуля ОбщегоНазначенияУТВызовСервера:
  - Функция ЗначенияРеквизитовОбъекта (Ссылка, Реквизиты) Экспорт
  - Функция ЗначениеРеквизитаОбъекта(Ссылка, ИмяРеквизита) Экспорт
  - Функция ЗначенияРеквизитовОбъектов (МассивСсылок, ИменаРеквизитов, ВыбратьРазрешенные = Ложь) Экспорт
  - Функция ЗначениеРеквизитаОбъектов (МассивСсылок, ИмяРеквизита, ВыбратьРазрешенные = Ложь) Экспорт
- Для выполнения кода на сервере следует использовать функцию ВыполнитьНаСервере (ПрограммныйКод).

  В параметр ПрограммныйКод следует передать программный код, который требуется выполнить на сервере.

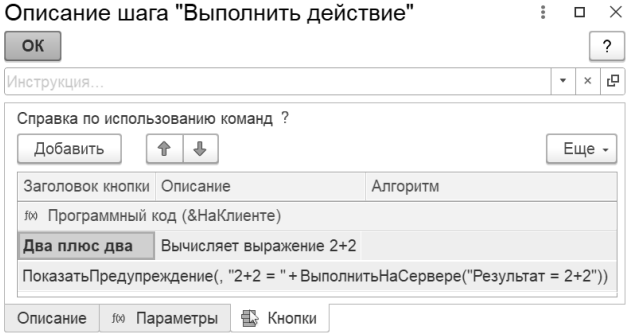
  Функция возвращает произвольное значение (переменная Результат).

  В функции доступен основной реквизит Объект (бип_Задача) и контекст формы задачи.

Пример

Пример с вызовом функции ВыполнитьНаСервере.

- В настройках сценария добавляется динамическая кнопка с выполняемым программным кодом:



- После запуска процесса по сценарию, создана задача, на форме которой отображается динамическая кнопка в виде гиперссылки:

Встроенный редактор программного кода

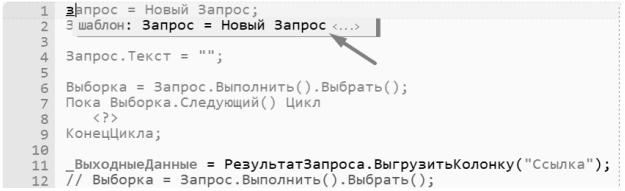
Общая информация

В Систему встроен редактор программного кода, который в первую очередь нацелен на облегчение чтения программного кода и написания несложных алгоритмов.

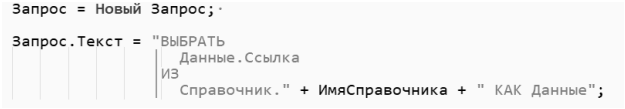
Возможности

- Использование индивидуальных *шаблонов программного кода*, доступных для быстрого выбора при редактировании программного кода.
  - При вводе символов, система предлагает список не только ключевых слов встроенного языка, но и готовых шаблонов.

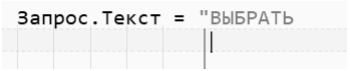
    Предварительный выбор такого шаблона отображается с просмотром всего текста шаблона, который встраивается в программный код (отображается серым цветом).
  - Для установки курсора в тексте программного кода после выбора шаблона, используется установка курсора (символы «<?>» в тексте шаблона).
  - Шаблоны могут быть настроены индивидуально для каждого пользователя и / или сразу для всех.



- Система отображает знаки табуляции и лишние проблемы в конце строк (красная точка).



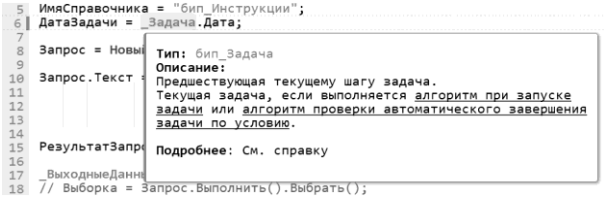
- Автоматическое добавление символа «|» в текстах запроса при переносе строки.



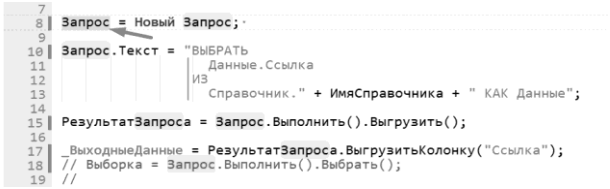
- Автоматическая установка пробелов после знаков «//».

```
// Алгоритм получения
// Может использоватьс
// В качестве примера
```

- Автоматическая установка знаков «;».
- Автоматические заглавные буквы для ключевых слов.
- Автоматическое добавление кавычек.
- Цветовое выделение переменных и функций Системы.
- Подсказки для функций переменных Системы.



- Подсветка встречаемости выделенного текста с отметкой строк, содержащих выделенный текст.



- Групповое комментирование / раскомментирование выделенного текста клавишами Shift + / .
- Тёмный режим.

Приложения

Пример программной настройки меню выбора для чат-ботов Telegram

Общая информация

Меню выбора может быть реализовано с помощью настроек справочника *Ключевые слова* 2 способами:

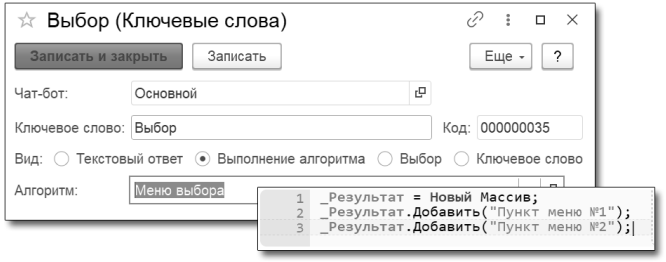
- настройкой ключевых слов с видом *Выбор*.
- настройка с помощью алгоритмов.

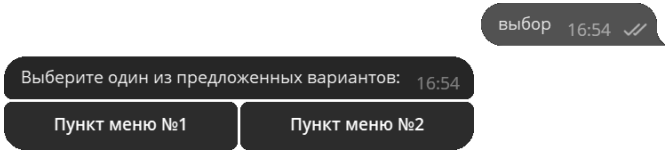
Смотрите Руководство пользователя «Сигнал» (раздел *Ключевые слова*).

Ниже будет рассмотрен пример настройки с помощью второго способа.

Настройка выбора

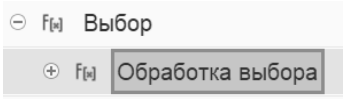
Если добавить новое ключевое слово с видом *Выполнение алгоритма* и в этом алгоритме в переменную _Результат передать массив, то элементы этого массива будет являться кнопками меню выбора.



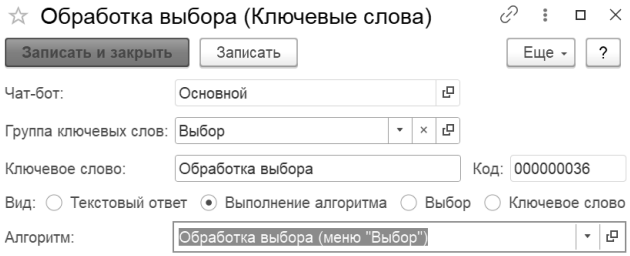


Настройка обработки выбора

Для обработки нажатия кнопок добавим новое ключевое слово, подчиненное ключевому слову *Выбор*.



Это ключевое слово будет, также, с видом *Выполнения алгоритма*.



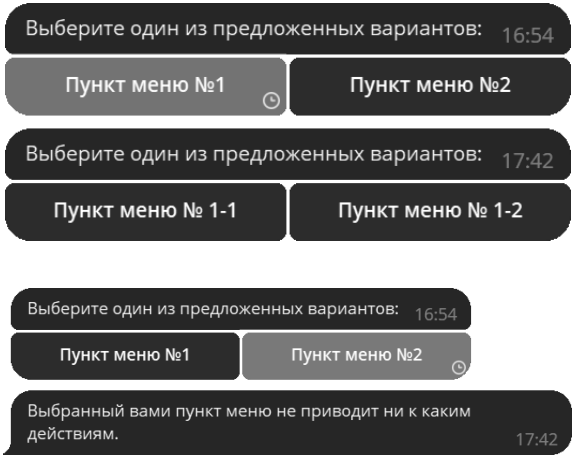
Для каждой кнопки определим значение переменной _Результат.

Для кнопки «*Пункт меню №1*» это будет новый *массив кнопок*.

Для кнопки «*Пункт меню №2*» это будет *сообщение* от чат-бота.

```
ТекущийОбработчикВыбора = _ДанныеДляОбработки.Получить("ТекущийОбработчикВыбора");
ВыбранноеЗначение       = _ДанныеДляОбработки[ТекущийОбработчикВыбора];

Если ВыбранноеЗначение = "Пункт меню №1" Тогда

    _Результат = Новый Массив;

    _Результат.Добавить("Пункт меню № 1-1");
    _Результат.Добавить("Пункт меню № 1-2")

ИначеЕсли ВыбранноеЗначение = "Пункт меню №2" Тогда

    _Результат = "Выбранный вами пункт меню не приводит ни к каким действиям."

КонецЕсли
```

В результате получим следующее взаимодействие в чат-боте:



Дальнейшая настройка

Для обработки подпунктов меню «*Пункт меню №1*» добавим новое ключевое слово, подчиненное предыдущему и настроем для него алгоритм обработки выбора.

В этот раз, вместо массива в переменную _Результат передадим *Структуру*, которая будет содержать следующие настройки:

- <Заголовок выбора> (тип Строка) – заголовок выбора пунктов меню.
- <Выбор> (тип Массив) – массив кнопок выбора.

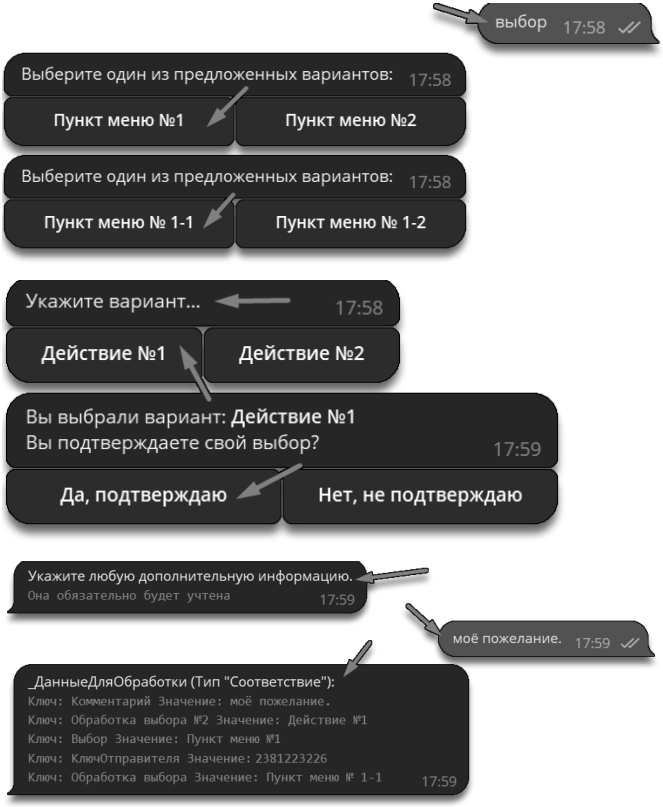
- <ПодтверждатьВыбор> (тип Булево) – признак подтверждения выбранного действия.
- <ЗаголовокКомментария> (тип Строка) – если указан, то чат-бот попросит отправить дополнительное сообщение с указанным заголовком.

```
ТекущийОбработчикВыбора = _ДанныеДляОбработки.Получить("ТекущийОбработчикВыбора");
ВыбранноеЗначение       = _ДанныеДляОбработки[ТекущийОбработчикВыбора];

Если ВыбранноеЗначение = "Пункт меню № 1-1" Тогда

   Выбор = Новый Массив;
   Выбор.Добавить("Действие №1");
   Выбор.Добавить("Действие №2");

   _Результат = Новый Структура;
   _Результат.Вставить("ЗаголовокВыбора",      "Укажите вариант...");
   _Результат.Вставить("Выбор",                Выбор);
   _Результат.Вставить("ПодтверждатьВыбор",    Истина);
   _Результат.Вставить("ЗаголовокКомментария", "Укажите любую дополнительную информацию.
                                               |<code>Она обязательно будет учтена</code>");

ИначеЕсли ВыбранноеЗначение = "Пункт меню № 1-2" Тогда

   _Результат = "Текущая дата: " + ТекущаяДата()

КонецЕсли
```

Для обработки нажатия кнопок «*Действие №1*» и «*Действие №2*» добавим алгоритм, который выведет в чат-бот «содержание» переменной _ДанныеДляОбработки (тип Соответствие).

Выбор	Меню выбора
Обработка выбора	Обработка выбора (меню "Выбор")
Обработка выбора №2	Обработка выбора №2 (меню "Выбор")
Вывод истории выбора	Вывод истории выбора

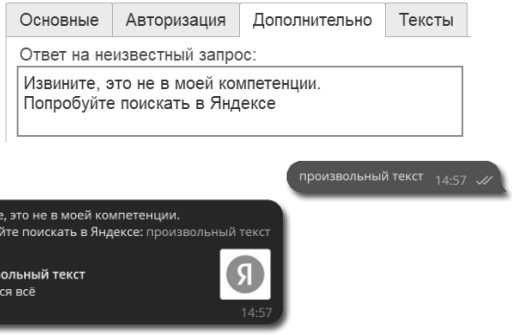
В переменной _ДанныеДляОбработки Система сохраняет всю историю выбора, которая может быть использована в алгоритмах для целей реализации реальных механизмов взаимодействия чат-ботов с пользователями.



Дополнительные возможности программной настройки чат-ботов Telegram

Обработка неизвестного запроса

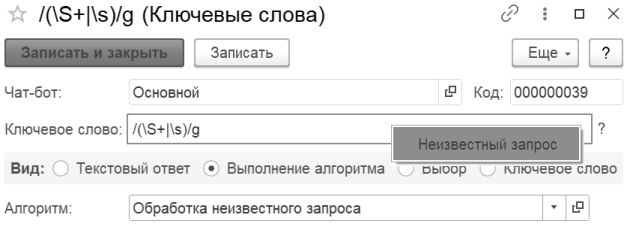
По умолчанию, при получении чат-ботом сообщения, для которого не предусмотрено ключевое слово, чат-бот ответит стандартной фразой, указанной в настройках чат-бота.



Для обработки таких «*неизвестных запросов*» можно использовать *Ключевое слово*.

Для этого достаточно выбрать в контекстном меню поля *Ключевое слово* пункт *Неизвестный запрос*.

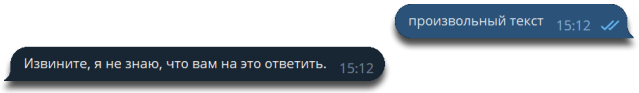
В поле *Ключевое слово* будет добавлена строка регулярного выражения.



В качестве примера можно настроить обработку неизвестного запроса с помощью алгоритма.

Самый простой алгоритм обработки неизвестного запроса может быть описан одной строкой.

```
1 _Результат = "Извините, я не знаю, что вам на это ответить."
```

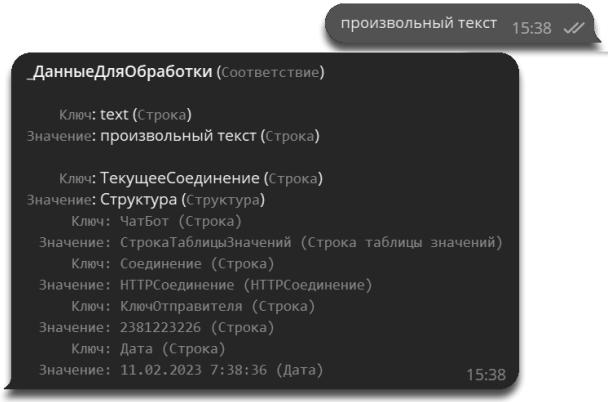


Чтобы получить *данные сообщения*, можно использовать переменную _ДанныеДляОбработки.

В качестве примера выведем содержимое переменной _ДанныеДляОбработки с помощью встроенной функции нс_Телеграм.ПолучитьДанныеБуфера.

```
_Результат = нс_Телеграм.ПолучитьДанныеБуфера(_ДанныеДляОбработки)
```

Переменная _ДанныеДляОбработки (тип Соответствие) может содержать разные значения, в зависимости от полученных данных.



Сохранение вложений

В переменной могут присутствовать поля, связанные с *вложениями*, присланными пользователем.

С помощью встроенной функции нс_Телеграм.СохранитьВложения можно сохранить полученные вложения в указанный каталог.

Подробнее, см. Процедуры и функции расширения «Сигнал».

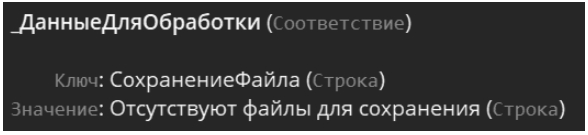
Система умеет сохранять вложения следующих типов:

- Фотография.
- Видео.
- Голосовое сообщение.
- Произвольный файл.

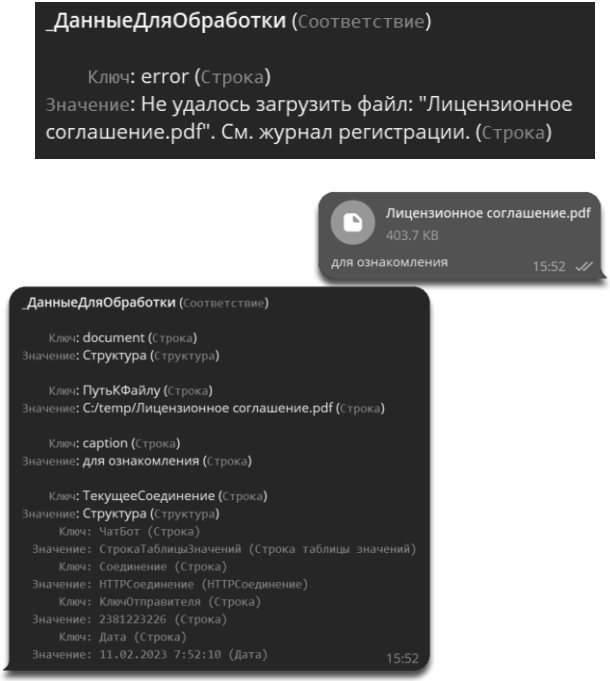
В качестве примера сохраним полученный от пользователя файл в папку «C:\temp\» и выведем содержимое переменной _ДанныеДляОбработки.

```
нс_Телеграм.СохранитьВложения(_ДанныеДляОбработки, "C:\Temp\");
_Результат = нс_Телеграм.ПолучитьДанныеБуфера(_ДанныеДляОбработки)
```

Перед выполнением сохранения Система предварительно проверяет, есть ли во входящем сообщении вложения.



При неудачной попытке сохранения в переменной _ДанныеДляОбработки будет добавлено значение с ключом <error>.



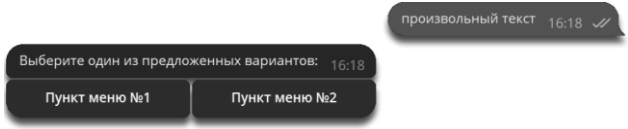
Переадресация обработки входящего сообщения

Если внутри алгоритма требуется *передать* обработку входящего сообщения другому обработчику ключевых слов, то для этого в переменную _ДанныеДляОбработки следует добавить ключ *Переадресация* и в качестве его значения передать *ссылку на ключевой слово*.

При этом, можно передать только ключевые слова 1-го уровня.

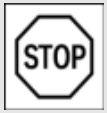
В качестве примера передадим обработку неизвестного запроса ключевому слову «*Выбор*», которое приводилось в качестве примера в разделе Пример программной настройки меню выбора для чат-ботов Telegram.

```
_ДанныеДляОбработки.Вставить(
"Переадресация", Справочники
.нс_КлючевыеСлова
.НайтиПоНаименованию("Выбор"));

_Результат = Истина
```



УВЕДОМЛЕНИЕ

Автор оставляет за собой право на опечатки, ошибки и технические изменения.



Копирование и распространение текста (всего или отдельных его частей) в любом виде и для любых целей без разрешения автора запрещено.

Все исключительные права на программный продукт
КонструкторБизнесПроцессов и документацию принадлежат
© ООО «Новые Системы», г. Иркутск, 2015–2025 г.
РосПатент: №2022667718 от 23.09.22
Единый реестр: №15564 от 18.11.22

Воспроизведение, полное или частичное, допускается только
с письменного разрешения автора.