


**ООО «Техноград плюс»**


**ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ  
ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД  
КТП ИЗ)**

**Руководство разработчика рабочих процессов**

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 2 из 45


## Оглавление

Термины и сокращения.....	3
1 Введение.....	4
2 Условия работы.....	5
3 Общая информация о рабочих процессах.....	6
3.1 Жизненный цикл.....	6
3.2 Категории.....	8
3.3 Служебный аргумент «Context».....	8
4 Управление рабочими процессами.....	10
4.1 Фазы выполнения.....	10
4.2 Построение и настройка.....	13
4.3 Вспомогательные операции.....	15
Приложение А. Библиотека активностей рабочего процесса.....	17
А.1 Управление.....	17
А.2 Выполнение.....	22
А.3 Примитивы.....	23
А.4 Транзакции.....	26
А.5 Коллекции.....	28
А.6 Обработка ошибок.....	31
А.7 Данные.....	31
Приложение Б. Описание пользовательского интерфейса.....	37
Б.1 Панель фильтрации.....	37
Б.2 Панель инструментов.....	38
Б.3 Область настройки рабочего процесса.....	38

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 3 из 45

## Термины и сокращения

Термин	Обозначение
ТЕХНОГРАД КТП ИЗ	ПО «ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ)» разработки ООО «Техноград плюс»
ТЕХНОГРАД ИПСД	ПО «ТЕХНОГРАД. ИНТЕГРАЦИОННАЯ ПЛАТФОРМА СЕРВИСОВ И ДАННЫХ (ТЕХНОГРАД ИПСД)» разработки ООО «Техноград плюс»
API	Application Program Interface (интерфейс программирования приложений)
ПИ	Пользовательский интерфейс
Заказ	Информационный объект внешней системы, содержащий набор услуг, которые заказал клиент для установки, а также коммерческие параметры этих услуг
Рабочий процесс	Последовательность активностей, которые должны быть выполнены для реализации услуги
Активность	Действие в рамках рабочего процесса, которое логически считается элементарным (неделимым)
Задание	Действие (или набор действий) или вызов API, который представлен как одна бизнес-транзакция и реализует логику взаимодействия с внешней системой
Услуга	Информационный объект, обозначающий потребности клиента во взаимодействии с поставщиком
Ресурс	Объект, имеющий под собой физическую логическую основу и представляющий собой средство, которое необходимо для оказания услуг клиенту (количество таких объектов обычно ограничено)

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 4 из 45


## 1 Введение

Данное руководство содержит описание основных задач, стоящих перед разработчиком рабочих процессов ПО (системы) «ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ)» (далее — «ТЕХНОГРАД КТП ИЗ»), и способах их решения.

Разработчиком рабочих процессов системы «ТЕХНОГРАД КТП ИЗ» могут быть выполнены следующие операции:


- Управление рабочими процессами, а именно:
  - создание,
  - настройка,
  - трассировка,
  - тестирование,
  - экспорт (импорт),
  - сравнение версий.
- Создание категорий рабочих процессов.
- Дополнительные операции, включая настройку необходимых справочников и определение пользовательских типов данных.

Информация о возможностях системы «ТЕХНОГРАД КТП ИЗ» и описание схемы ее работы приведено в документе *Руководство администратора «ТЕХНОГРАД КТП ИЗ»*.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 5 из 45

## 2 Условия работы

Для обеспечения возможности работы с пользовательским интерфейсом ПО «ТЕХНОГРАД КТП ИЗ» необходимо, чтобы на используемом компьютере был установлен браузер, реализованный на движке Chromium и используемая учетная запись обладала правами и привилегиями, необходимыми для запуска и разработки рабочих процессов.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 6 из 45

### 3 Общая информация о рабочих процессах

Рабочий процесс в рамках ПО «ТЕХНОГРАД КТП ИЗ» представляет собой законченный фрагмент бизнес-логики, который может состоять из активностей различного уровня сложности, а именно:

- Простой активности (элементарное действие, которое логически считается неделимым).
- Набора простых активностей.
- Активности типа «Задание» (**Task**).
- Активности типа «Процесс», содержащей ранее реализованный рабочий процесс.

Построение рабочих процессов осуществляется в пользовательском интерфейсе ПО «ТЕХНОГРАД КТП ИЗ» с помощью конструктора рабочих процессов.

**Примечание.** 1. Описание элементов пользовательского интерфейса ПО «ТЕХНОГРАД КТП ИЗ», необходимых для построения рабочих процессов, приведено в Приложении Б.  
 2. Описание процедуры построения и настройки рабочих процессов приведено в разделе 4.

Для построения рабочих процессов могут использоваться настраиваемая библиотека активностей, которая представляет собой ранее созданные рабочие процессы, а также системные активности, разработанные ООО «Техноград плюс».

**Примечание.** Описание стандартных активностей, содержащихся в ПО «ТЕХНОГРАД КТП ИЗ», приведено в приложении А.

#### 3.1 Жизненный цикл

Жизненный цикл рабочего процесса начинается с момента его запуска и заканчивается после завершения его обработки.

Активности, которые содержит рабочий процесс, выполняются сверху вниз (см. рисунок 1). Порядок выполнения рабочего процесса может быть изменен при помощи активностей из раздела **Управление**.

**Примечание.** Описание активностей раздела **Управление** приведено в приложении А.

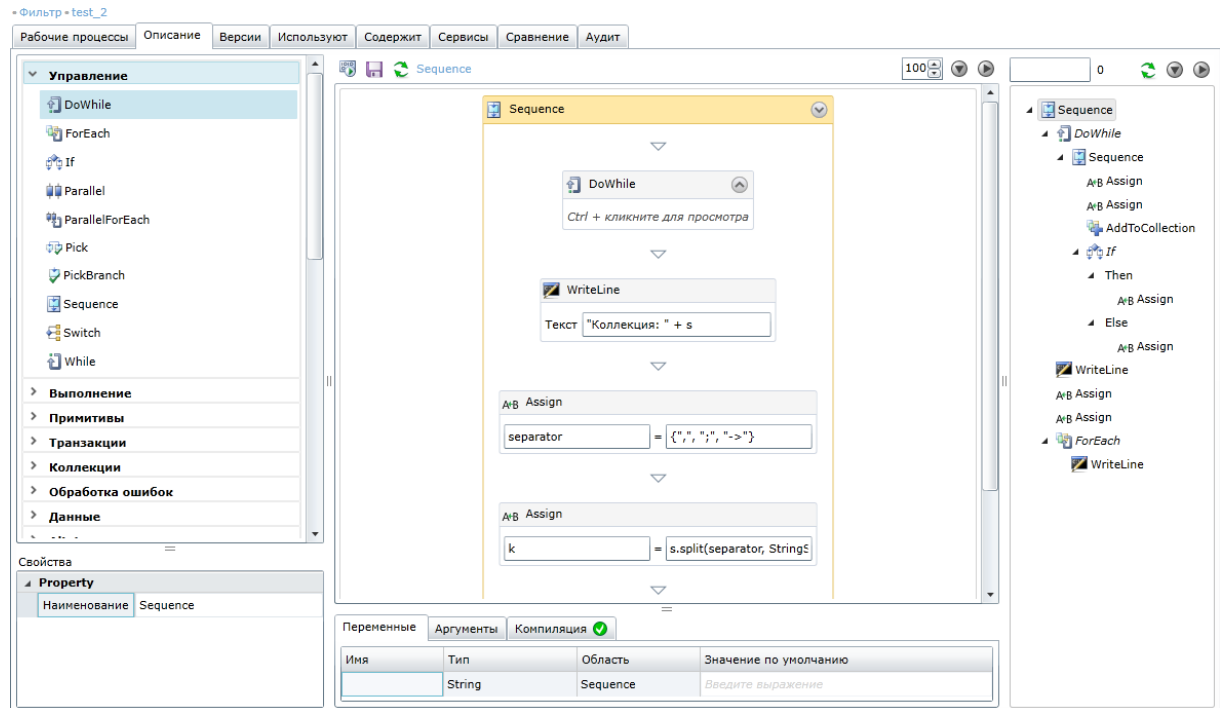


Рисунок 1 — Конструктор рабочего процесса

Если рабочий процесс не завершен, то он может иметь один из следующих статусов:


- **Создание** — рабочий процесс находится в процессе создания.  
При этом состоянии среда выполнения не обрабатывает рабочий процесс.
- **Готов к выполнению** — рабочий процесс готов к обработке.  
При этом статусе среда выполнения обрабатывает объект автоматически.
- **Выполнение** — рабочий процесс находится на стадии обработки средой выполнения.
- **Ошибка** — обработка рабочего процесса остановлена с ошибкой.

После завершения обработки рабочий процесс может иметь следующие статусы:

- **Выполнен** — обработка рабочего процесса завершена успешно.
- **Отменен** — обработка рабочего процесса отменена пользователем.
- **Прерван** — обработка рабочего процесса прервана.

Выполнение рабочего процесса может быть остановлено до его естественного завершения. Причиной такого завершения рабочего процесса может быть один из следующих вариантов:

- Отмена выполнения рабочего процесса.  
Причиной могут быть следующие ситуации:
  - Отмена выполнения рабочего процесса пользователем.
  - Выполнение необработанных исключений.
- Остановка (прерывание) выполнения рабочего процесса, выгрузка его из памяти с возможностью возобновления выполнения с места последнего его сохранения.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 8 из 45

Такое сохранение рабочего процесса реализуется при помощи активности **Persist**.

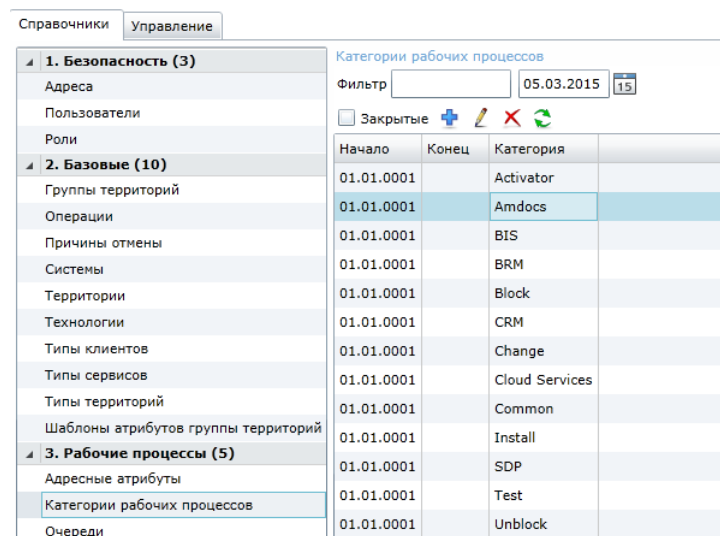
Прерывание рабочего процесса может возникнуть в следующих случаях:

- При выполнении действий (исключений), определенных в активностях из раздела **Обработка ошибок** или активности **Task**.
- Возникновение ошибок в процессе выполнения активности **DatumNode** (при установке свойства **AbortInstanceOnFailure**).
- При прерывании процесса по инициативе пользователя.
- При возникновении ошибки инфраструктуры (не доступен сервер интегратора внешних систем, недоступен сервер баз данных и т. д.).

## 3.2 Категории

В системе «ТЕХНОГРАД КТП ИЗ» предусмотрена возможность логически сгруппировать рабочие процессы. Для этого используется справочник **Категории рабочих процессов** (см. рисунок 2).

Элементы данного справочника используются в конструкторе рабочих процессов для группировки активностей.



Начало	Конец	Категория
01.01.0001		Activator
01.01.0001		Amdocs
01.01.0001		BIS
01.01.0001		BRM
01.01.0001		Block
01.01.0001		CRM
01.01.0001		Change
01.01.0001		Cloud Services
01.01.0001		Common
01.01.0001		Install
01.01.0001		SDP
01.01.0001		Test
01.01.0001		Unblock


Рисунок 2 — Справочник «Категории рабочих процессов»

## 3.3 Служебный аргумент «Context»

Аргумент **Context** предназначен для получения и преобразования данных из рабочего процесса. Свойства и процедуры данного аргумента, которые могут использоваться при построении рабочих процессов, приведены в таблицах 1 и 2.

Если при объявлении аргументов какого-либо рабочего процесса область их видимости установлена как **Context**, то данные атрибуты могут быть доступны для всех рабочих процессов внутри исполняемого процесса.




	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 9 из 45

**Таблица 1 — Свойства аргумента «Context»**

Свойство	Назначение (тип)
WorkflowInstanceId	идентификатор запущенного на выполнение экземпляра рабочего процесса (число)
Debug	признак отладки рабочего процесса (логический)
DebugConnection	источник строки подключения к интегратору сервисов данных в режиме отладки (строка)
DebugCommandPrefix	ссылка на источник данных в интеграторе сервисов данных в режиме отладки (строка)

**Таблица 2 — Процедуры аргумента «Context»**

Процедура	Назначение (тип)
Serialize (<Значение>)	преобразование <Значение> в последовательность формата XML (строка)
Deserialize (of <Тип>) (<Значение>)	преобразование <Значение> из последовательности формата XML в указанный тип данных <Тип>
UpdateTaskAttr (<Идентификатор задания>, <Наименование атрибута>, <Значение атрибута>)	обновление атрибута задания (логический)
CancelTask (<идентификатор задания>)	отмена задания (логический)
CloseTask (<идентификатор задания>)	закрытие задания (логический)

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 10 из 45

## 4 Управление рабочими процессами

### 4.1 Фазы выполнения

Реализация какого-либо бизнес-процесса требует определение через один или несколько рабочих процессов.

Выполнение рабочих процессов осуществляется некоторыми этапами (фазами). Каждый рабочий процесс всегда содержит хотя бы одну фазу, которая определена неявно — фаза транзакции, после которой выполнение заказа отменить невозможно.

Рабочий процесс, который используется для реализации, рекомендуется создавать с использованием следующих фаз:

- **Check** — фаза, отделяющая проверку входных параметров процесса.
- **Install** — фаза, отделяющая выполнение дополнительных действий для реализации бизнес-процесса (например, бронирование или выделение ресурсов, определение технической возможности и т. п.).

При необходимости количество фаз может быть изменено, но при этом все рабочие процессы, исполняемые в рамках одного заказа, должны содержать одинаковое количество фаз.

**Примечание.** Правила настройки фаз описаны в п. 4.1.1.

Для создания фазы в рабочем процессе используется активность **Task**, у которой установлен флажок **Pulse**. Каждая такая активность обозначает одну фазу, а выполнение этой активности — окончание фазы.

Если в заказе присутствует несколько рабочих процессов, то их обработка (см. рисунок 3; зависимость процессов последовательная) будет происходить в следующем порядке:

1. Запуск на выполнение получает первый рабочий процесс (согласно графу рабочих процессов). Осуществляется проверка входных параметров. После чего начинается выполнение фазы **Check** первого процесса. При этом выполнение первого процесса приостанавливается и осуществляется передача на запуск второго рабочего процесса.
2. Для второго рабочего процесса также осуществляется проверка входных параметров и выполнение фазы **Check**. После чего выполнение второго процесса приостанавливается и осуществляется запуск третьего процесса.
3. Для третьего рабочего процесса также осуществляется проверка входных параметров и выполнение фазы **Check**. После чего осуществляется закрытие фазы **Check** первого рабочего процесса. При этом выполнение третьего рабочего процесса приостанавливается.
4. Выполнение фаз **Install** и транзакции осуществляется аналогично.

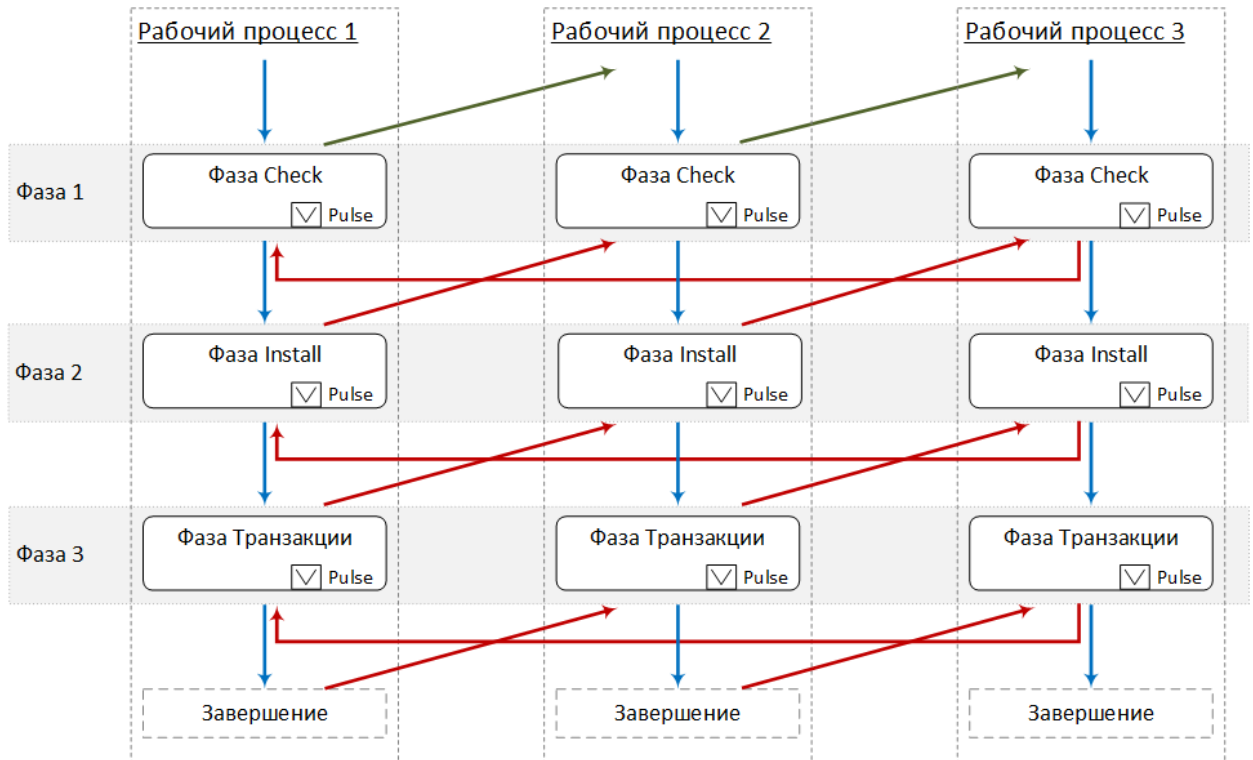


Рисунок 3 — Порядок выполнения фаз в рабочих процессах

5. После запуска фазы транзакции запускаются обработчики подтверждений, содержащиеся в активности **CompensableActivity** (если такие были в теле рабочего процесса). На этом этапе отмена выполнения рабочего процесса становится невозможной.
6. Если ошибок при выполнении рабочих процессов не обнаружено, то заказ считается выполненным.

**Примечание.** На рисунке 3 используются следующие способы выделения стрелок:

- синий цвет — выполнение действий согласно сценарию рабочего процесса;
- зеленый цвет — запуск рабочего процесса согласно графу рабочих процессов;
- красный цвет — закрытие фазы (закрытие активности **Task**, обозначающей фазу).

Отмена изменений в заказе происходит в обратном порядке выполнения рабочих процессов. Если один из процессов завершился с ошибкой, то произойдет отмена изменений, выполненных ошибочным рабочим процессом, а также произойдет отмена результатов всех процессов в этом заказе.

#### 4.1.1 Ограничения

Настройка фаз рабочих процессов должна выполняться с учетом следующих правил:

- Состав фаз во всех рабочих процессах ветки должен совпадать.

**Правильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 2
Фаза 3	Фаза 3	Фаза 3

**Неправильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 2
Фаза 4	Фаза 3	Фаза 3

**Неправильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 2
	Фаза 3	Фаза 3

**Неправильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
	Фаза 1	Фаза 1
	Фаза 2	Фаза 2
	Фаза 3	Фаза 3

— Порядок фаз во всех рабочих процессах ветки должен совпадать.

**Правильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 2
Фаза 3	Фаза 3	Фаза 3

**Неправильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 3	Фаза 2
Фаза 3	Фаза 2	Фаза 3

— В рамках одного рабочего процесса названия фаз должны быть уникальны.

**Правильно**


Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 2
Фаза 3	Фаза 3	Фаза 3

**Неправильно**

Рабочий процесс 1	Рабочий процесс 2	Рабочий процесс 3
Фаза 1	Фаза 1	Фаза 1
Фаза 2	Фаза 2	Фаза 1
Фаза 1	Фаза 2	Фаза 3

## 4.2 Построение и настройка

Для построения и настройки рабочего процесса необходимо выполнить следующие действия:

1. Перейти на вкладку **Процессы**.
2. Щелкнуть кнопку , расположенную на панели инструментов.
3. Указать в открывшемся окне **Добавить** параметры рабочего процесса (название, категорию и дату начала действия). Щелкнуть кнопку **Сохранить**.
4. Перейти на вкладку **Описание**, расположенную в области настройки рабочего процесса (см. рисунок 4).

**Примечание.** Описание пользовательского интерфейса ПО «ТЕХНОГРАД КТП ИЗ», необходимого для построения и настройки рабочих процессов, приведено в приложении Б.

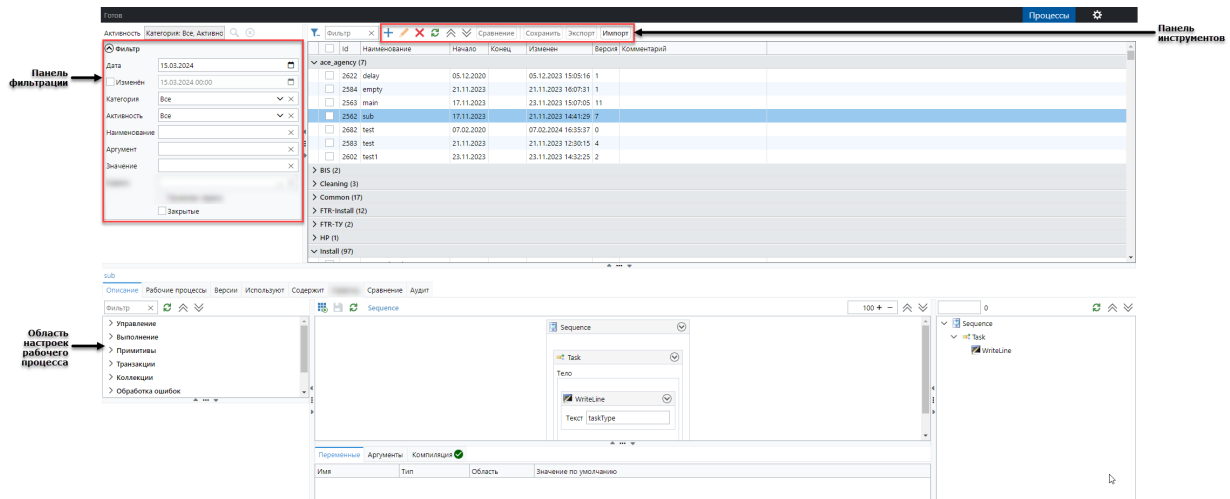



Рисунок 4 — Вкладка «Процессы»

5. Добавить в рабочей области в активность **Sequence**, установленную по умолчанию, две активности **Task** (для обозначения фаз **Check** и **Install**) путем перетаскивания их из библиотеки активностей (см. рисунок 5). Для каждой из активностей **Task** выполнить следующие действия:
  - а) Указать в поле **Наименование** названия активностей (фаза **Check** и фаза **Install**).
  - б) Установить флажок **Pulse**.
  - в) Указать в поле **Name** название фазы (**Check** и **Install**).
  - г) Указать в поле **Note** значение "".
  - д) Установить, при необходимости, остальные свойства активности.
6. Добавить в рабочую область остальные активности, выбрав их из библиотеки, и определить их свойства.

**Примечание.** Описание активностей приведено в приложении А.

7. Указать, при необходимости, переменные и атрибуты.

8. Щелкнуть кнопку  для сохранения рабочего процесса.

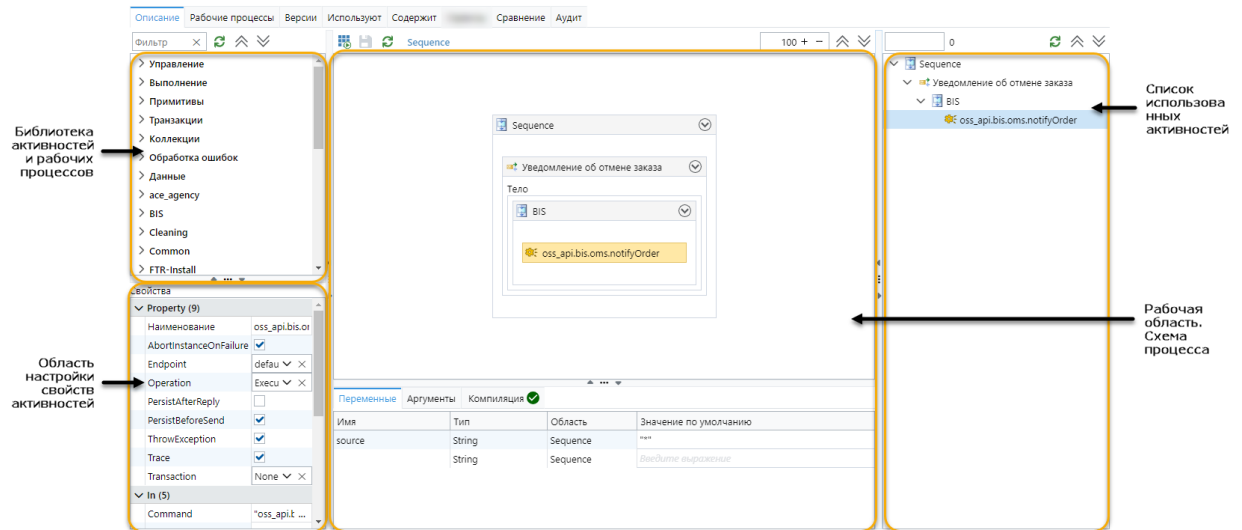






Рисунок 5 — Область настройки рабочего процесса. Вкладка «Описание»

9. Щелкнуть кнопку  для компиляции.

**Примечание.** При успешной компиляции процесса на вкладке **Компиляция** отображается значок , в противном случае —  с указанием ошибок.

10. Если ошибок при настройке рабочего процесса не обнаружено, перейти на вкладку **Рабочие процессы**.
11. Щелкнуть кнопку  для запуска рабочего процесса и проверки корректности его выполнения в рамках поставленной задачи.
12. Если для запуска рабочего процесса необходимо указать входные значения атрибутов, то ввести их значения в открывшемся окне **Запустить рабочий процесс** (см. рисунок 6) в полях, соответствующих названиям атрибутов, и щелкнуть кнопку **Сохранить**.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 15 из 45

**Запустить рабочий процесс** [Fullscreen] [Close]

Дата:  [Calendar] [Info]


Рабочий процесс:  [Info]

Комментарий:  [Close] [Info]

Атрибуты

Атрибут	Тип	Направление	Область	Значение	
OrderStatus	String	In	None	"COMPLETED"	... X
orderResultCode	String	In	None		... X
callbackEndpoint	String	In	Context		... X
orderResultText	String	In	None		... X

Рисунок 6 — Окно «Запустить рабочий процесс»

13. При возникновении ошибок в процессе выполнения рабочего процесса щелкнуть кнопку . Перейти на вкладку **Трассировка** и выяснить, где возникла ошибка, пошагово проверив каждое действие, выполненное рабочим процессом.

**Примечание.** Описание режима трассировки приведено в п. Б.3.1.1.

## 4.3 Вспомогательные операции

Для настройки рабочих процессов могут использоваться следующие справочники:

- **Категории рабочих процессов** — для группировки рабочих процессов.
- **Типы аргументов** — для задания новых типов аргументов или параметров рабочих процессов.
- **Типы заданий** — для определения типов заданий.


**Примечание.** Тип задания может быть задан при построении рабочего процесса в активности **Task** в свойстве **Name**.

### 4.3.1 Определение типов аргументов

Типы данных можно условно разделить на два вида:

- *Системные* — типы данных, используемые в .Net Framework.
- *Пользовательские* — созданные пользователем собственные типы данных, описываемые при помощи схемы XSD.

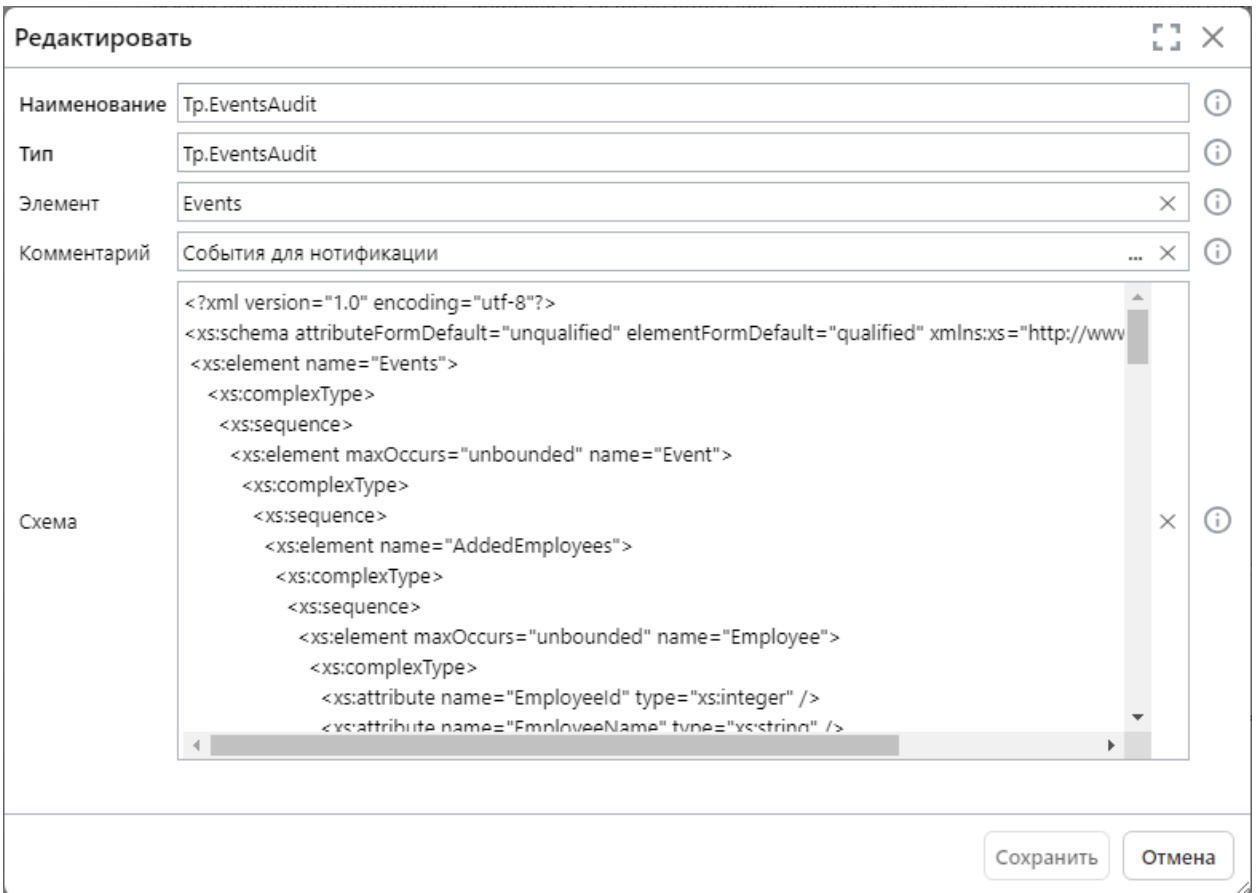
Аргументы такого типа можно передавать рабочему процессу в формате XML и сериализовать (десериализовать) их в переменные пользовательского типа.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 16 из 45

Для задания нового типа данных необходимо настроить справочник **Типы аргументов** (см. рисунок 7), элемент которого может иметь следующие свойства:

- **Наименование** — краткое название типа данных.
- **Тип** — полное название типа данных.
- **Элемент** — название основного типа данных, в случае если в XSD-схеме описаны несколько типов данных.
- **Комментарий** — дополнительная информация.
- **Схема** — описание XSD-схемы.

После этого в конструкторе рабочих процессов будет доступен тип данных с именем, указанным в поле **Наименование**.



Redactirovat

Наименование:

Тип:

Элемент:


Комментарий:

Схема: 

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www
<xs:element name="Events">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="Event">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AddedEmployees">
              <xs:complexType>
                <xs:sequence>
                  <xs:element maxOccurs="unbounded" name="Employee">
                    <xs:complexType>
                      <xs:attribute name="EmployeeId" type="xs:integer" />
                      <xs:attribute name="EmployeeName" type="xs:string" />
```

Рисунок 7 — Окно «Добавить» для добавления элемента в справочник «Типы аргументов»



	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 17 из 45

## Приложение А. Библиотека активностей рабочего процесса

### А.1 Управление

#### А.1.1 Активность «DoWhile»

Активность **DoWhile** (см. рисунок 8) предназначена для многократного выполнения вложенной активности, пока остается истинным заданное условие. Выполняется хотя бы один раз.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Condition** — для указания условия выполнения.

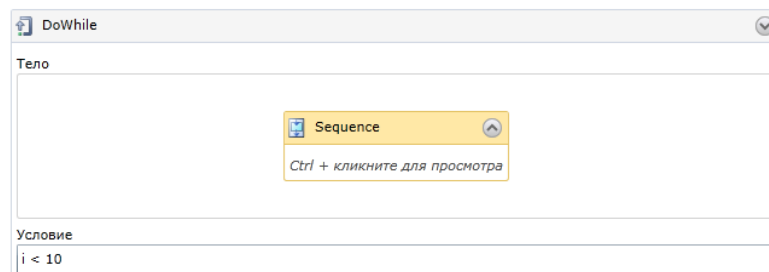


Рисунок 8 — Пример применения активности «DoWhile»

#### А.1.2 Активность «ForEach»

Активность **ForEach** (см. рисунок 9) предназначена для выполнения вложенных активностей один раз для каждого элемента коллекции, заданной в свойстве **Values**.

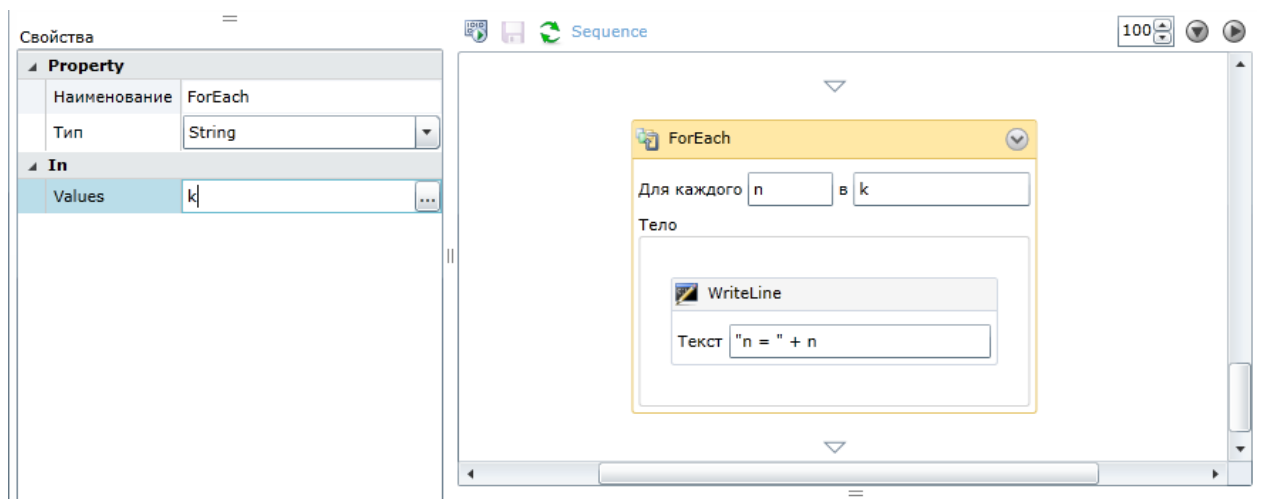


Рисунок 9 — Пример применения активности «ForEach»

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа элементов коллекции.

- **Values** — для указания переменной, обозначающей коллекцию.

### A.1.3 Активность «If»

Активность **If** (см. рисунок 10) предназначена для выполнения одной из двух вложенных активностей. Выполняется первая вложенная активность, если свойство **Condition** принимает истинное значение, вторая — если значение ложно.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Condition** — для указания условия выполнения.

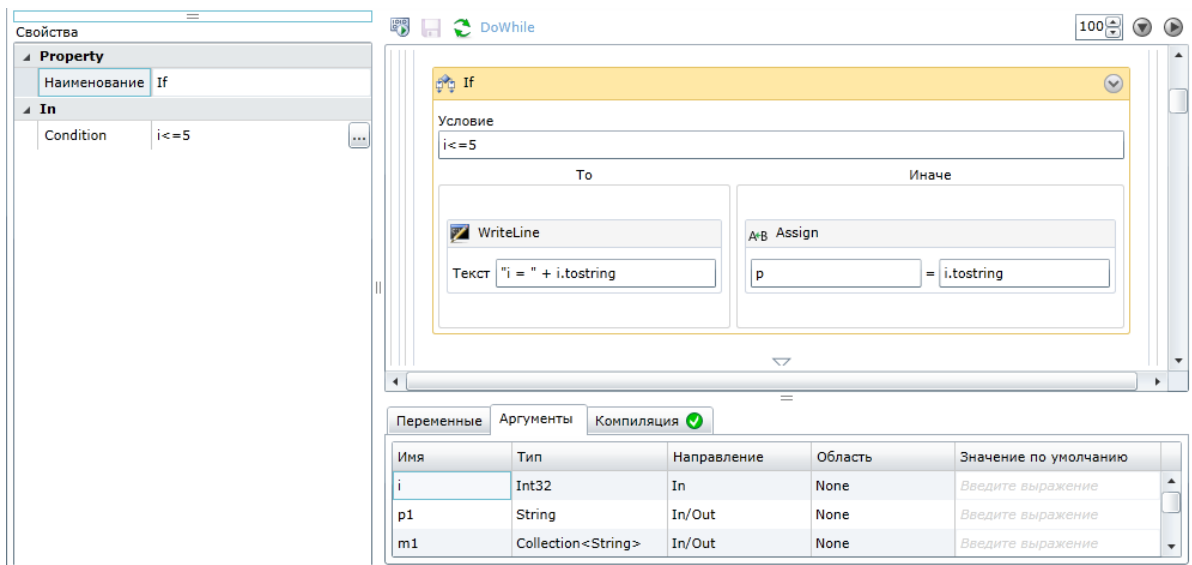


Рисунок 10 — Пример применения активности «If»


### A.1.4 Активность «Parallel»

Активность **Parallel** (см. рисунок 11) предназначена для выполнения всех вложенных активностей одновременно (параллельно) и асинхронно. Выполнение активности завершается, когда все вложенные активности завершены или когда свойство

**CompletionCondition** принимает истинное значение.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **CompletionCondition** — для указания условия выполнения.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 19 из 45

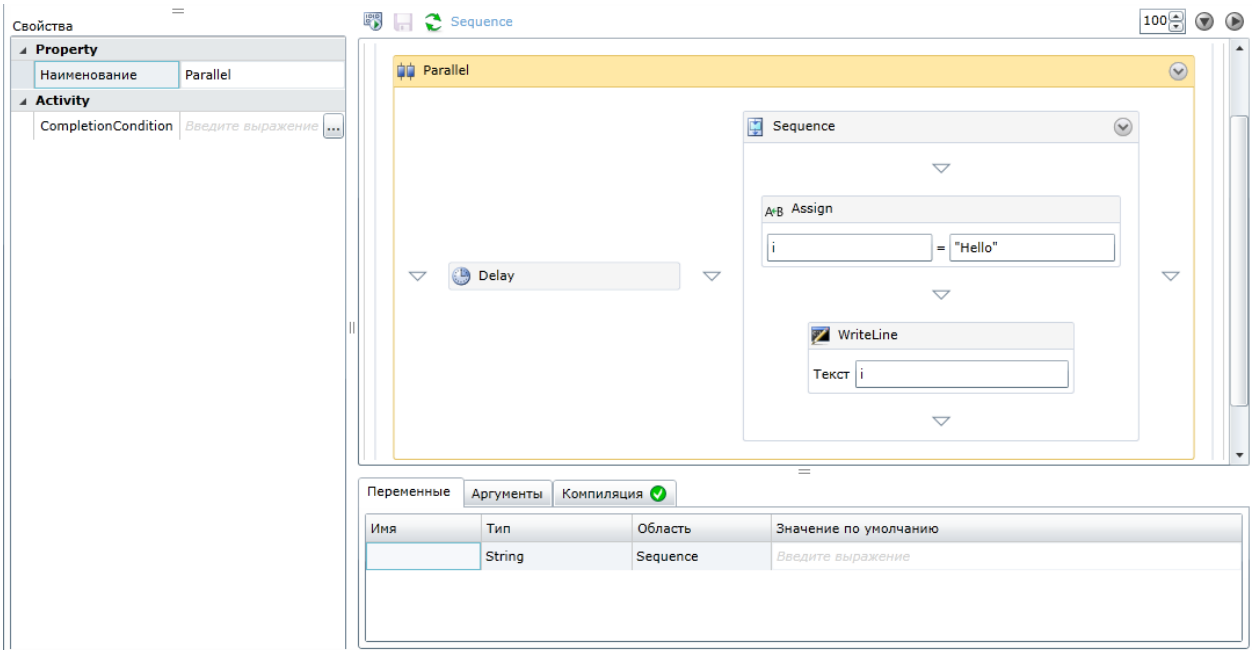


Рисунок 11 — Пример применения активности «Parallel»

### A.1.5 Активность «ParallelForEach»

Активность **ParallelForEach** (см. рисунок 12) предназначена для параллельного выполнения вложенных активностей для каждого элемента коллекции, которая задана в свойстве **Values**.

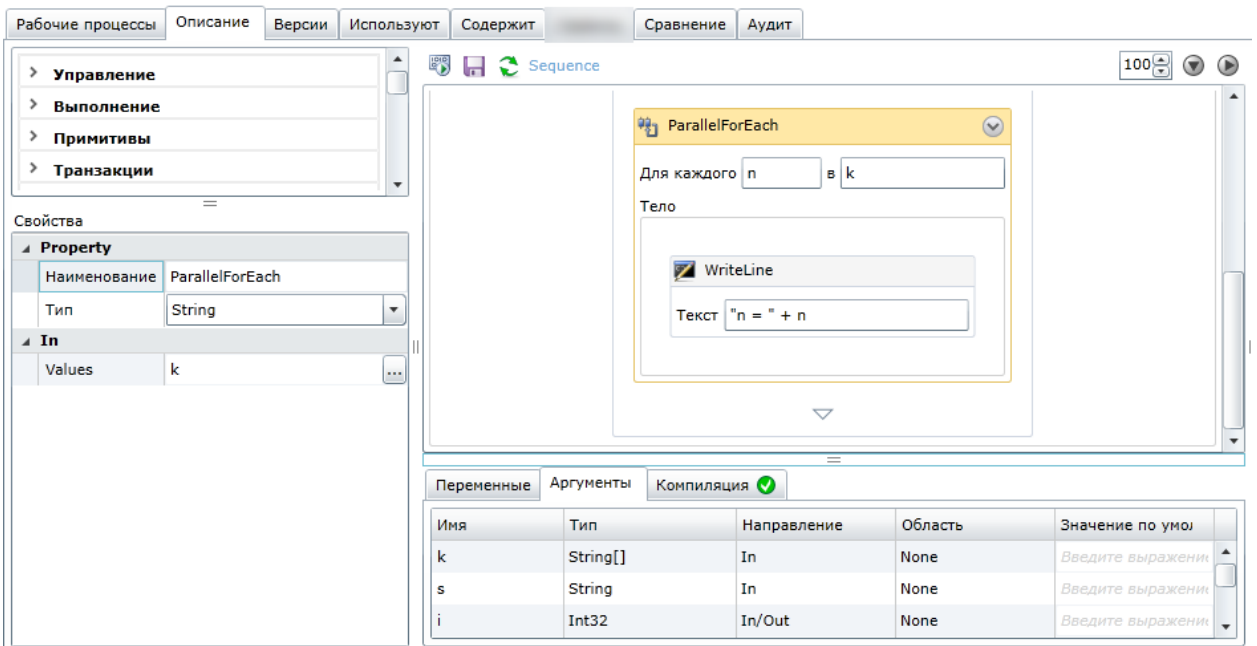



Рисунок 12 — Пример применения активности «ParallelForEach»

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 20 из 45

- **Тип** — для указания типа элементов коллекции.
- **Values** — для указания коллекции.

### A.1.6 Активности «Pick» и «PickBranch»

Активности **Pick** и **PickBranch** (см. рисунок 13) используются совместно. Активность **Pick** предназначена для обеспечения моделирования потока управления на основе событий. Активность **Pick** выполняет только одну из ветвей, включающую активность **PickBranch**.

**PickBranch** — потенциальный путь выполнения активности **Pick**. Активность **PickBranch** включает в себя два элемента: **Триггер** и **Действие**.

При запуске активности **Pick** выполняются все активности, содержащиеся в элементах **Триггер** всех **PickBranch**. Если корректно выполняется первый триггер и выполняется соответствующее действие, то все другие активности остальных триггеров отменяются.

В примере, приведенном на рисунке 13, после выполнения всех активностей на экран будет выведен результат: Hello.

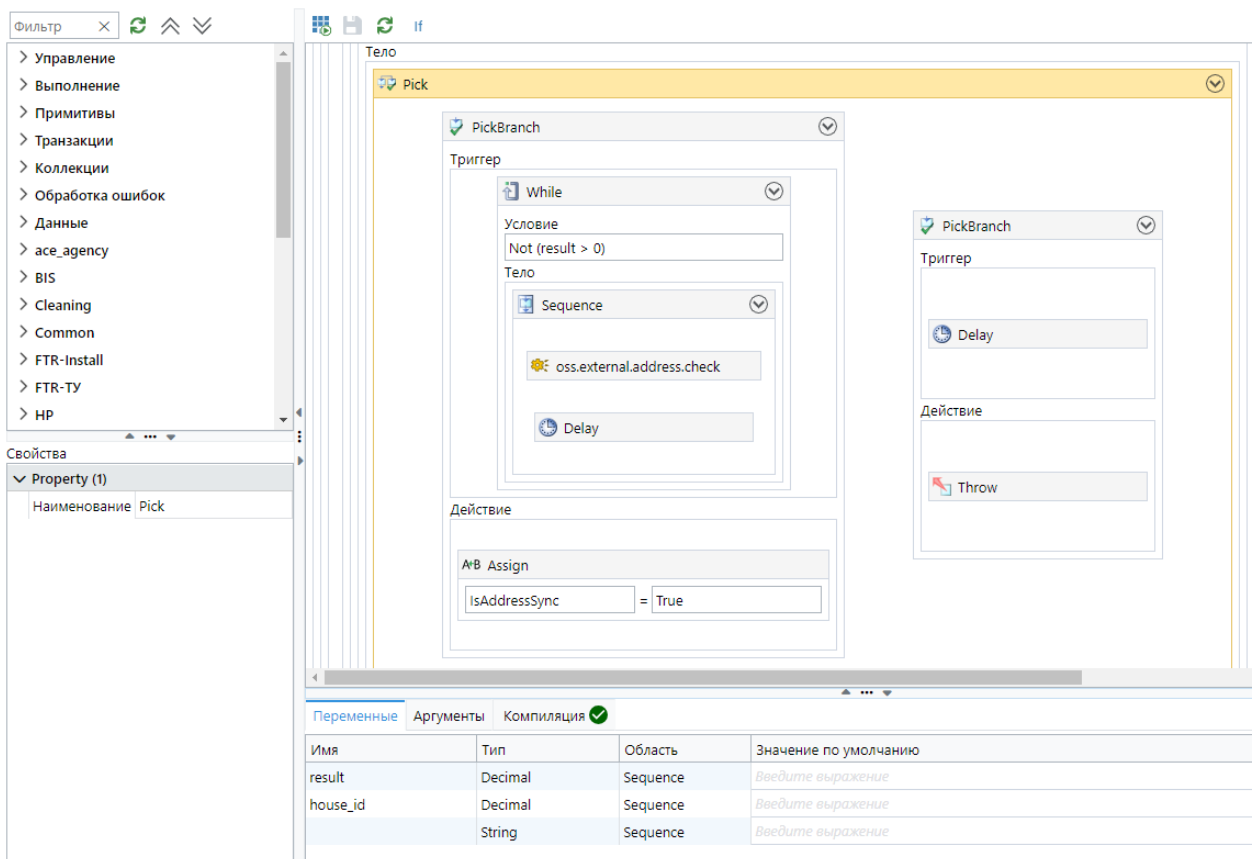



Рисунок 13 — Пример применения активностей «Pick» и «PickBranch»

### A.1.7 Активность «Sequence»

Активность **Sequence** предназначена для выполнения вложенных активностей в соответствии с заданным порядком. Данная активность может содержать любое количество других активностей, которые будут выполняться в порядке их расположения сверху вниз.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 21 из 45

**Примечание.** При создании любого рабочего процесса рекомендуется использовать активность **Sequence**, в которую следует добавлять требуемые активности в нужном порядке.

### A.1.8 Активность «Switch»

Активность **Switch** (см. рисунок 14) предназначена для выполнения одной из вложенных активностей в зависимости от значения выражения в свойстве **Expression**. Результат вычисления выражения сравнивается с вариантами значений (варианты значений должны быть различны): если обнаружено совпадение, то выполняется соответствующая активность.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа элементов коллекции.
- **Expression** — для указания выражения.

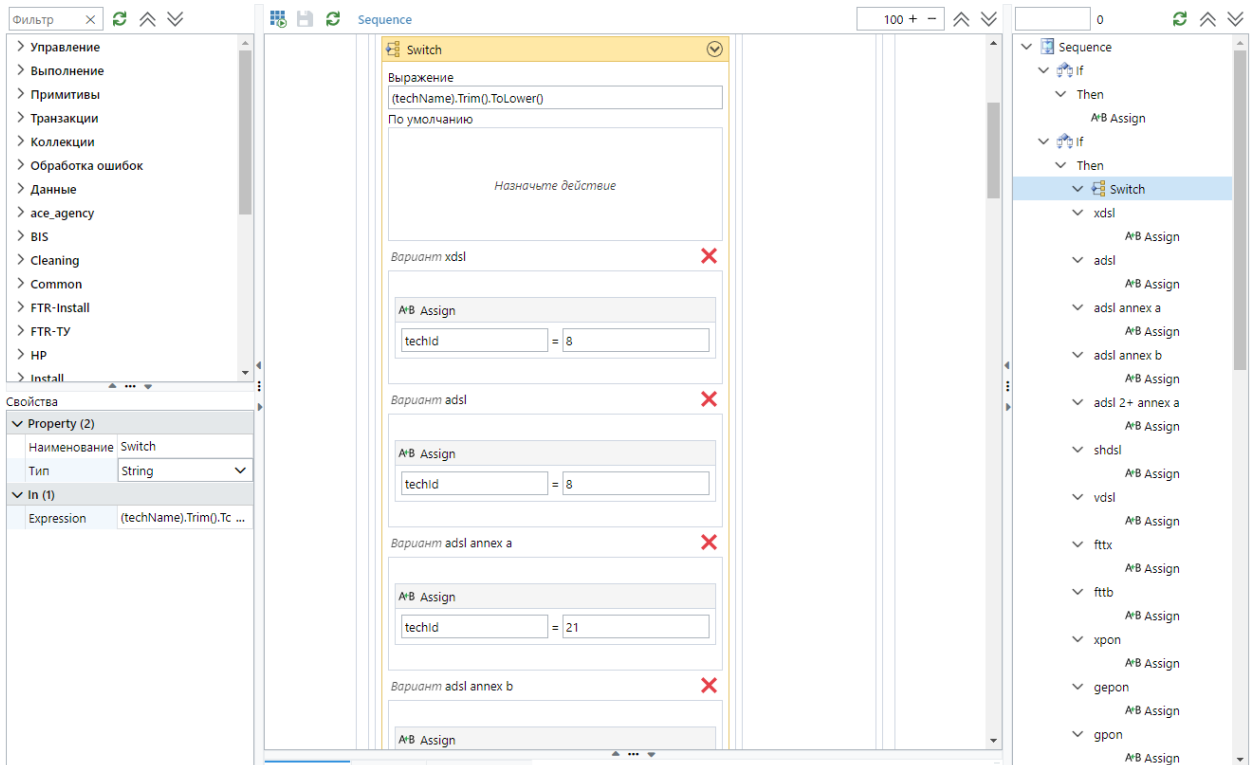


Рисунок 14 — Пример применения активности «Switch»

### A.1.9 Активность «While»

Активность **While** (см. рисунок 15) предназначена для многократного выполнения вложенных активностей, пока остается истинным заданное условие. Перед каждой итерацией вычисляется значение свойства **Condition**. Если результат вычисления значения оказывается ложным, то выполнение активности сразу прекращается.

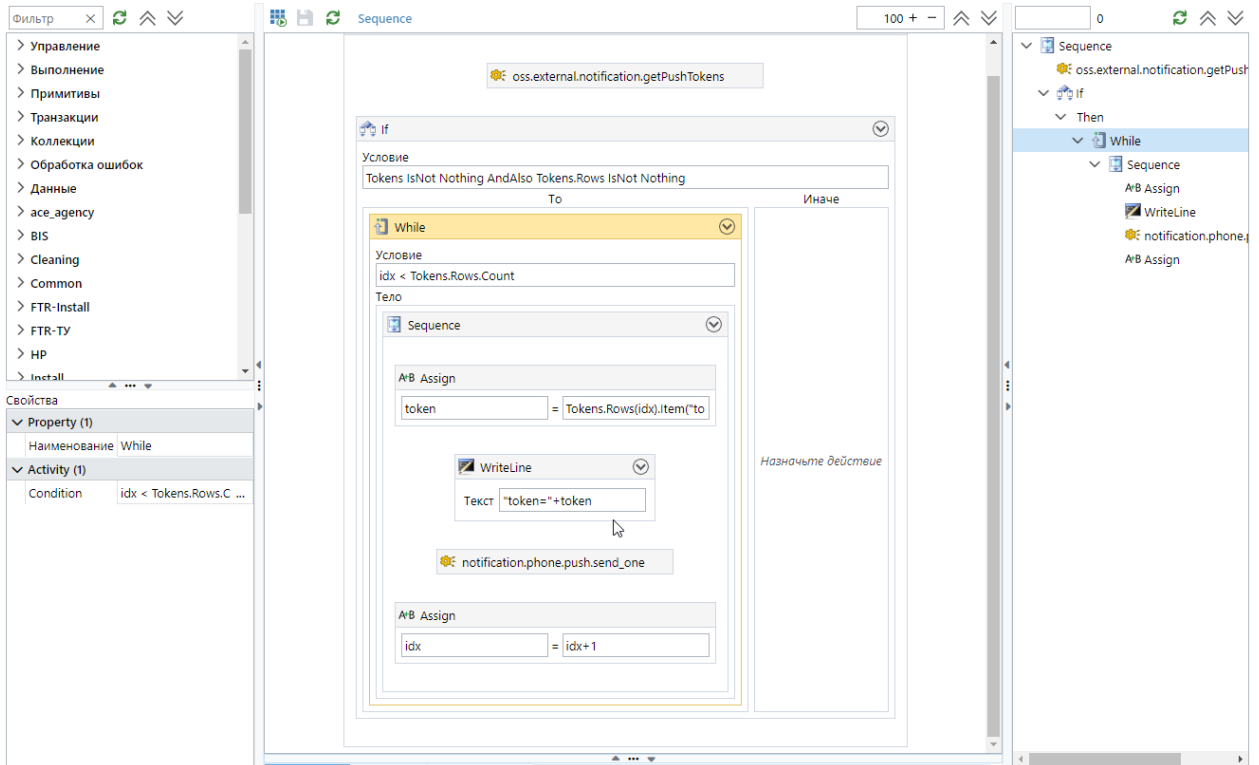


Рисунок 15 — Пример применения активности «While»

## А.2 Выполнение

### А.2.1 Активность «Merge»

Активность **Merge** в текущей версии программы не используется.

### А.2.2 Активность «NoTraceScore»

Активность **NoTraceScore** в текущей версии программы не используется.

### А.2.3 Активность «Persist»

Активность **Persist** предназначена для сохранения экземпляра рабочего процесса, название которого указывается в свойстве **Наименование**.


### А.2.4 Активность «TerminateWorkflow»

Активность **TerminateWorkflow** предназначена для завершения выполнения экземпляра рабочего процесса.

После завершения рабочий процесс не может быть возобновлен.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Exception** — для указания исключения, спровоцировавшее завершение рабочего процесса.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 23 из 45

— **Reason** — для указания причины завершения выполнения рабочего процесса.

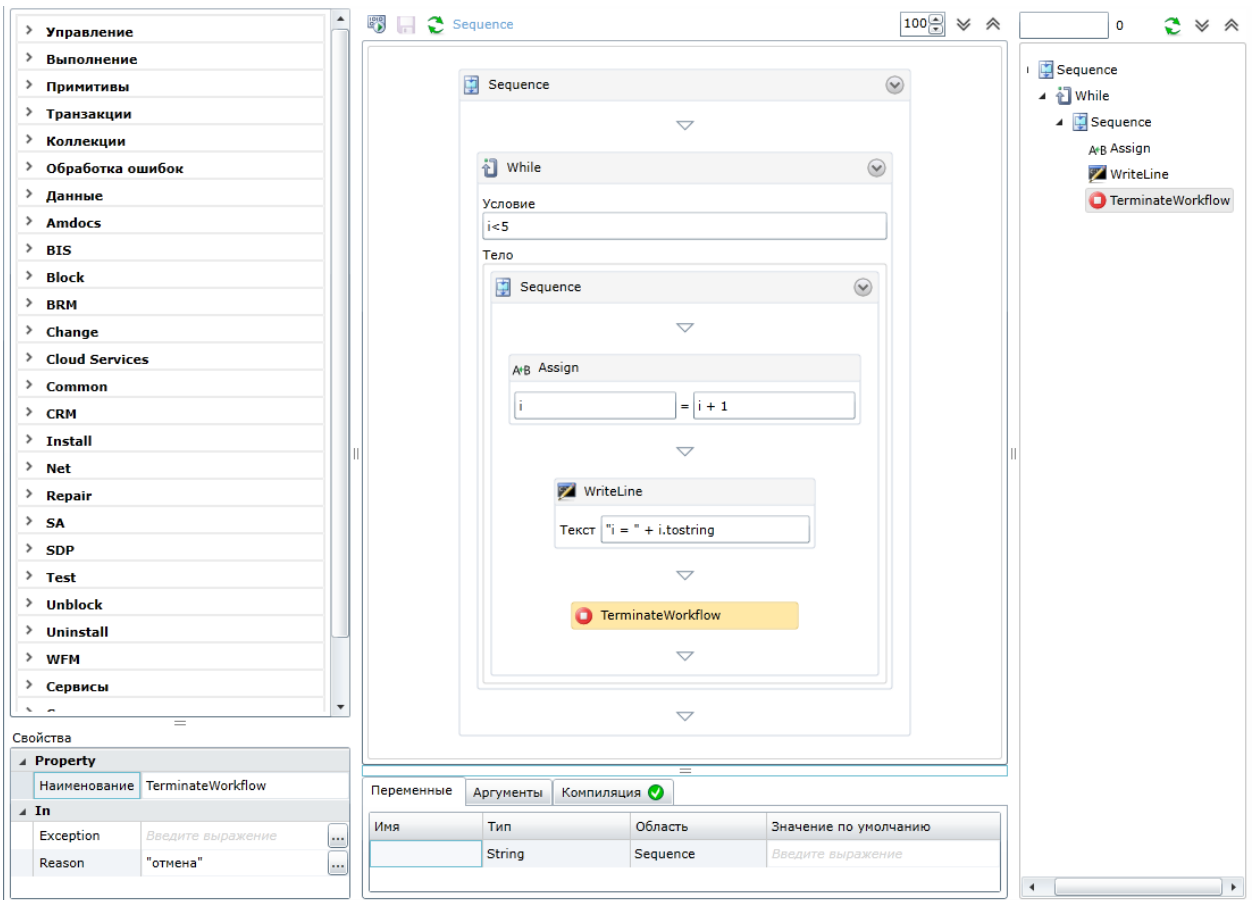


Рисунок 16 — Пример применения активности «TerminateWorkflow»

Результатами выполнения рабочего процесса будут следующие данные:

- рабочий процесс будет иметь статус **Отменен**;
- результат одного пройденного цикла активности **While** (например, при введенном значении 0 аргумента  $i$  на экране будет выведено:  $i = 1$ );
- причина отмены рабочего процесса, указанная в поле **Reason**.

## А.3 Примитивы

### А.3.1 Активность «Delay»

Активность **Delay** предназначена для задания временного интервала задержки выполнения какого-либо действия.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Duration** — для указания длительности интервала.

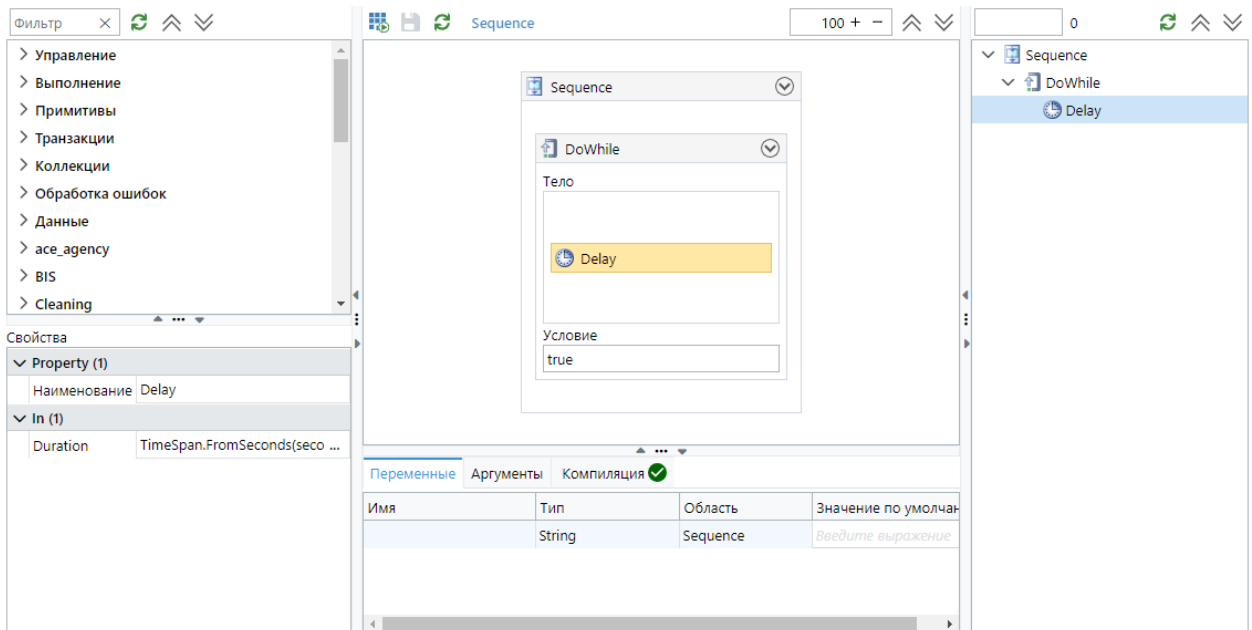


Рисунок 17 — Пример применения активности «Delay»

### А.3.2 Активность «Script»

Активность **Script** предназначена для задания сценария. Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Code** — для указания текста сценария, написанного в виде программного кода на C#.
- **Parameters** — для указания списка необходимых параметров с помощью одноименного окна.

### А.3.3 Активность «Assign»

Активность **Assign** (см. рисунок 18) предназначена для установки значения аргументов в пределах рабочего процесса.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа аргумента.
- **To** — для указания аргумента.
- **Value** — для указания значения (выражения) аргумента.



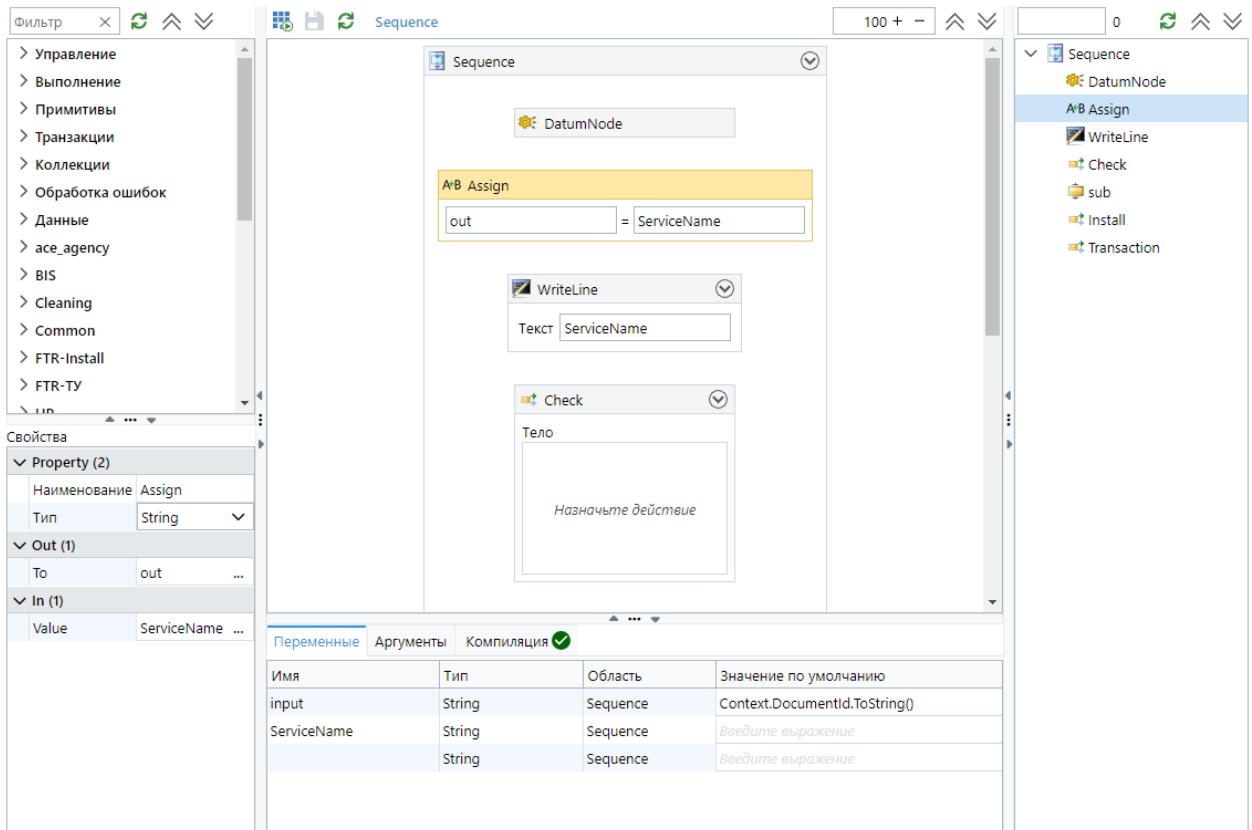



Рисунок 18 — Пример применения активности «Assign»

### А.3.4 Активность «InvokeMethod»

Активность **InvokeMethod** предназначена для вызова метода для указанного объекта или типа.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа возвращаемого значения.
- **MethodName** — для указания названия метода, который будет вызван при выполнении активности.
- **RunAsynchronously** — для включения (выключения) режима, при котором указанный метод будет вызван асинхронно.
- **TargetType** — для указания типа объекта, в котором содержится заданный метод.
- **GenericTypeArguments** — для указания обобщенных типов аргументов.
- **Parameters** — для указания необходимых параметров заданного метода.
- **Result** — для указания результата выполнения метода.
- **TargetObject** — для указания объекта источника метода.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 26 из 45

### А.3.5 Активность «WriteLine»

Активность **WriteLine** (см. рисунок 14) предназначена для записи заданного значения в выходной поток, а также для проверки корректного выполнения процесса или активности.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Text** — для указания строки для записи.
- **TextWriter** — для указания объекта-приемника строки.

## А.4 Транзакции

### А.4.1 Активность «CancellationScope»

Активность **CancellationScope** содержит два раздела, которые определяют ее назначение, а именно:

- Раздел **Тело**, в который помещаются активности, составляющие логику действия.
- Раздел **Обработчик**, в который помещаются активности, обеспечивающие логику отмены действия.

Отменено может быть только незавершенное действие. Если запрос отмены запланирован и активности в разделе **Тело** не завершились, то активность **CancellationScope** будет отмечена как **Canceled**, после чего будут выполнены активности в разделе **Обработчик**.

### А.4.2 Активность «CompensableActivity»


Активность **CompensableActivity** позволяет отменить, подтвердить или компенсировать выполненные ранее активности.

Данная активность определяется следующими разделами:

- **Тело** — содержит активности, которые выполняют неделимые элементы бизнес логики.
- **Обработчик компенсации** — содержит активности, которые будут выполняться в случае отмены рабочего процесса после успешного завершения выполнения всех активностей, заданных в разделе **Тело**.
- **Обработчик подтверждения** — содержит активности, которые будут выполняться на фазе завершения транзакции в случае успешного завершения активностей, заданных в разделе **Тело**.
- **Обработчик отмены** — содержит активности, которые будут выполняться в случае отмены рабочего процесса до успешного завершения выполнения всех активностей, заданных в разделе **Тело**.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Result** — для указания результирующего аргумента выполнения активности.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 27 из 45

#### А.4.3 Активность «Compensate»

Активность **Compensate** предназначена для явного вызова обработчика компенсации, заданного в активности **CompensableActivity**.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Target** — для указания аргумента, который должен быть использован для вызова обработчика компенсации.

**Примечание.** Аргумент, заданный в свойстве **Target**, должен иметь тип **CompensationToken**.

#### А.4.4 Активность «Confirm»

Активность **Confirm** предназначена для явного вызова обработчика подтверждения, заданного в активности **CompensableActivity**.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Target** — для указания аргумента, который должен быть использован для вызова обработчика подтверждения.

**Примечание.** Аргумент, заданный в свойстве **Target**, должен иметь тип **CompensationToken**.

#### А.4.5 Активность «TransactionScope»

Активность **TransactionScope** предназначена для обозначения границ блока рабочего процесса, принимающего участие в транзакции.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **AbortInstanceOnTransactionFailure** — для включения (выключения) режима, обозначающего, следует ли прерывать рабочий процесс при прерывании транзакции.
- **IsolationLevel** — для указания степени изоляции транзакции.

Степени изоляции могут быть следующих типов:

- **Chaos** — режим, когда данные (с более высоким приоритетом), измененные в транзакциях, не могут быть перезаписаны.
- **ReadCommitted** — режим, когда во время обработки транзакции данные могут считываться и изменяться.
- **ReadUncommitted** — режим, когда во время обработки транзакции данные не могут считываться и изменяться.
- **RepeatableRead** — во время обработки транзакции данные могут считываться, но не могут изменяться. В ходе выполнения транзакции возможно добавление новых данных.
- **Serializable** — режим, когда во время обработки транзакции данные могут считываться, но не могут изменяться. Добавление новых данных невозможно.

- **Snapshot** — режим, когда данные могут считываться. Перед тем, как транзакция изменяет данные, она проверяет, не изменены ли данные другой транзакцией после их первоначального считывания. Если данные обновлены, то возникает ошибка. Это позволяет транзакции получить значение данных, зафиксированное ранее.
  - **Unspecified** — режим, когда могут использоваться различные степени изоляции, но степень не может быть заранее определена.
- **Timeout** — для указания времени ожидания транзакции.

## А.5 Коллекции

### А.5.1 Активность «AddToCollection»

Активность **AddToCollection** (см. рисунок 19) предназначена для добавления элемента в указанную коллекцию.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа элементов в коллекции.
- **Collection** — для указания коллекции, в которую добавляется новый элемент.
- **Item** — для указания элемента для добавления в коллекцию.

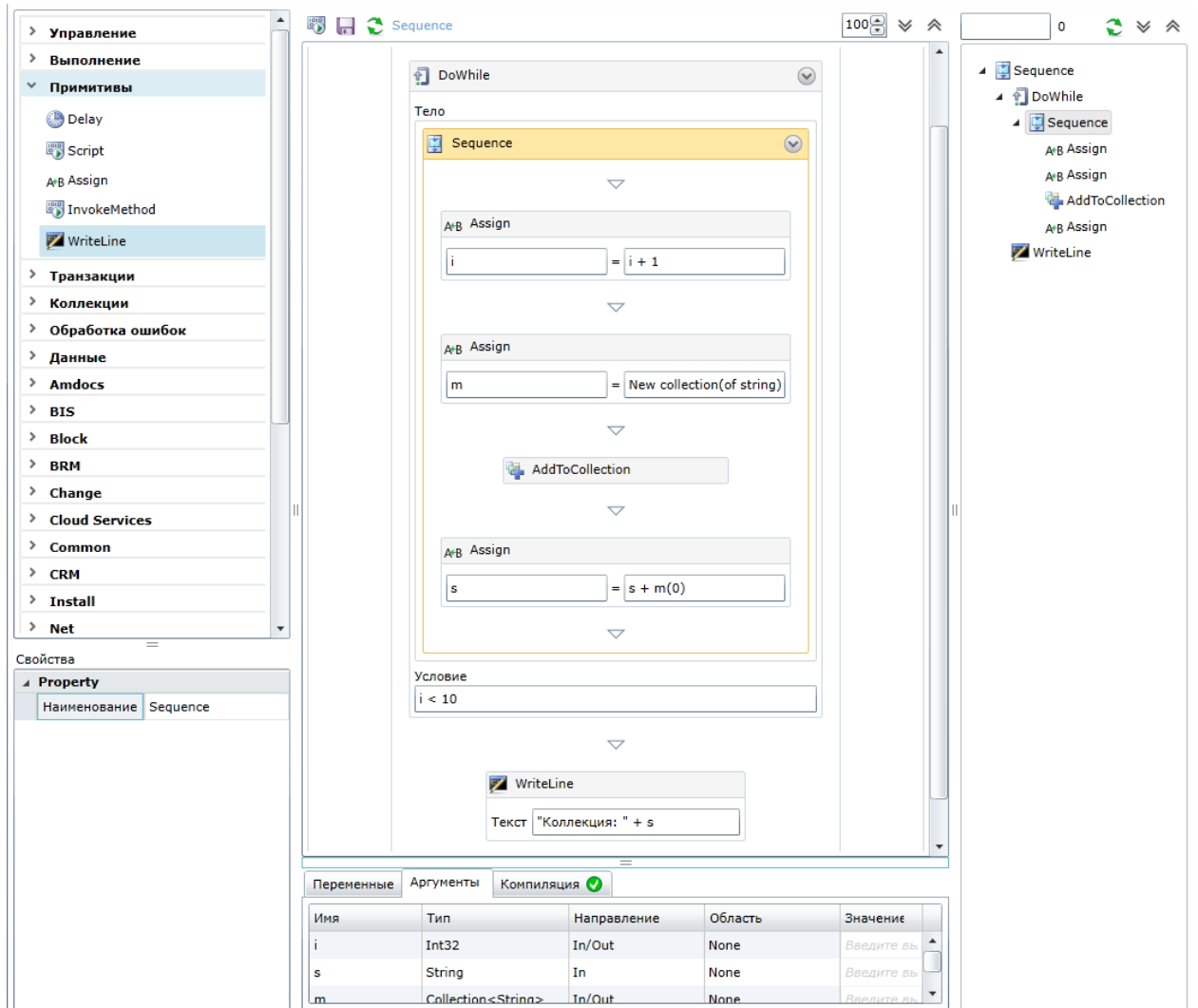


Рисунок 19 — Пример применения активности «AddToCollection»

### A.5.2 Активность «ClearCollection»

Активность **ClearCollection** предназначена для удаления элементов из указанной коллекции.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа элементов в коллекции.
- **Collection** — для указания коллекции, из которой требуется удалить элементы.

### A.5.3 Активность «ExistsInCollection»

Активность **ExistsInCollection** предназначена для проверки наличия требуемого элемента в указанной коллекции.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.

- **Тип** — для указания типа элементов в коллекции.
- **Collection** — для указания коллекции, в которой требуется найти элемент.
- **Item** — для указания элемента, который требуется найти в коллекции.
- **Result** — для получения результата проверки наличия элемента.

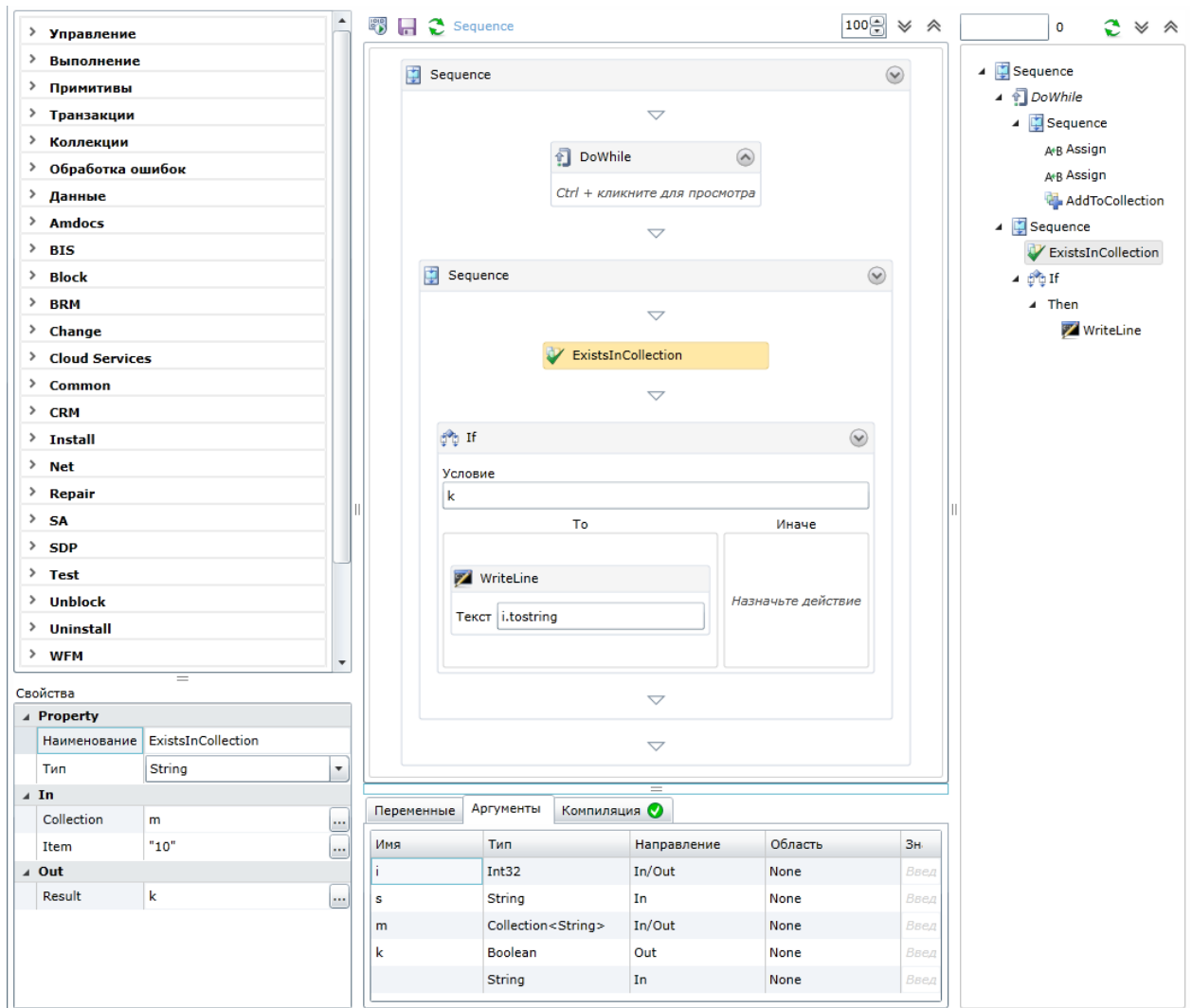



Рисунок 20 — Пример применения активности «ExistInCollection»

#### A.5.4 Активность «RemoveFromCollection»

Активность **RemoveFromCollection** предназначена для удаления указанного элемента из коллекции.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Тип** — для указания типа элементов в коллекции.
- **Collection** — для указания коллекции, из которой требуется удалить элемент.
- **Item** — для указания элемента, который требуется удалить из коллекции.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 31 из 45

## А.6 Обработка ошибок

### А.6.1 Активность «Throw»

Активность **Throw** предназначена для создания исключения (аномальной ситуации). Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **Exception** — для указания исключения, вызванного активностью.

### А.6.2 Активность «Rethrow»

Активность **Rethrow** предназначена для вызова ранее созданного исключения из раздела **Обработки** активности **TryCatch**.

### А.6.3 Активность «TryCatch»

Активность **TryCatch** содержит три раздела, которые определяют ее назначение, а именно:

- Раздел **Попытка**, в который помещаются активности, содержащие исключения. Действия этого раздела выполняются в первую очередь.
- Раздел **Обработки**, в который помещается коллекция исключений. Активности этого раздела выполняются, если действия из раздела **Попытка** спровоцировали вызов исключения. Далее это исключение сравнивается с элементами коллекции в порядке их добавления. Если совпадение найдено, то для этого исключения будет выполнен соответствующий блок обработки.
- Раздел **Завершение**, в который помещаются активности, выполняемые после того как все активности **TryCatch** выполнятся.


## А.7 Данные

### А.7.1 Активность «DatumNode»

Активность **DatumNode** (см. рисунок 21) предназначена для выполнения команды, опубликованной в ПО «ТЕХНОГРАД. ИНТЕГРАЦИОННАЯ ПЛАТФОРМА СЕРВИСОВ И ДАННЫХ (ТЕХНОГРАД ИПСД)».

Для активности **DatumNode** доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **AbortInstanceOnFailure** — для управления поведением активности при возникновении ошибки при ее выполнении. Если данный флажок установлен, то при возникновении ошибки в ходе выполнения активности **DatumNode** процесс будет продолжен; в противном случае — процесс прервется.
- **EndPoint** — для указания адреса **DatumNode**.
- **Operation** — для указания типа операции для команды **DatumNode**. Операции могут быть следующих типов:

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 32 из 45

- ExecuteBatch** — тип операции, когда команда выполняется для каждого элемента массива параметров.  
 Все входные и выходные параметры должны быть заданы как массивы данных (если команда принимает параметр типа `Int`, то необходимо задать `Int[]`). Результат выполнения команды вернется в виде массива результатов.
- ExecuteNonQuery** — тип операции, когда команда выполнится без возврата результирующего набора данных.  
 Все выходные параметры, сопоставляемые с параметрами команды, заполняются данными.
- ExecuteQuery** — тип операции, когда результат выполнения команды будет представлен в формате XML в свойстве **XML**.
- ExecuteReader** — тип операции, когда результат выполнения команды будет представлен в виде табличных данных в свойстве **Table**.
- ExecuteScalar** — тип операции, когда результатом выполнения команды является первый столбец первой строки результирующего набора, возвращаемого запросом. Дополнительные столбцы и строки игнорируются.

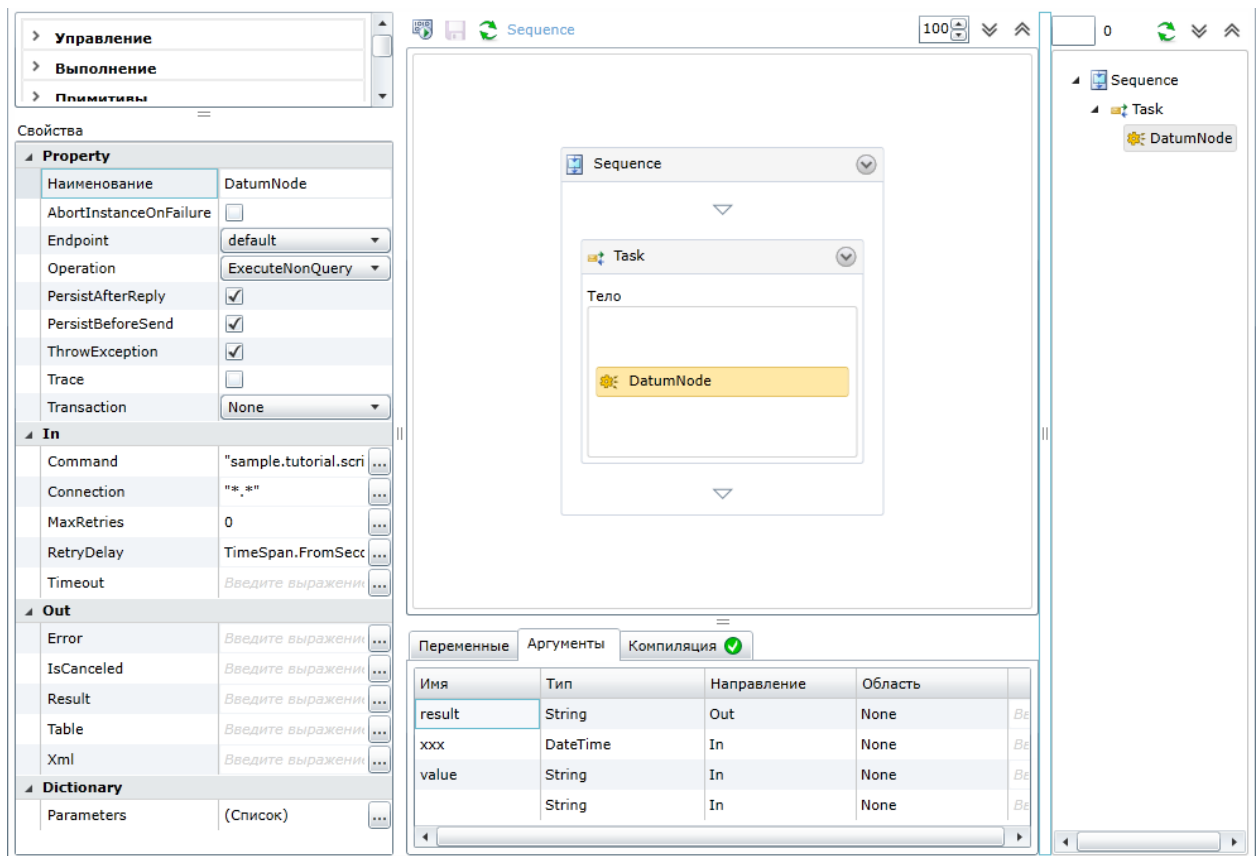



Рисунок 21 — Пример применения активности «DatumNode»

- PersistAfterReply** — для включения (выключения) режима, при котором определяется необходимость сохранения состояния экземпляра рабочего процесса после выполнения команды.



	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 33 из 45

- **PersistBeforeReply** — для включения (выключения) режима, при котором определяется необходимость сохранения состояния экземпляра рабочего процесса перед выполнением команды. В случае возникновения ошибки в ходе выполнения рабочего процесса он будет продолжен с момента последнего сохранения (в случае если флажки **AbortInstanceOnFailure** и **ThrowException** не установлены).
- **ThrowException** — для включения (выключения) режима, при котором определяется необходимость продолжения выполнения рабочего процесса при возникновении ошибки в команде.
- **Trace** — для включения (выключения) режима трассировки выполнения команды. Результат трассировки размещен в разделе **DatumNode** свойств экземпляра рабочего процесса.
- **Transaction** — для указания типа транзакции. Данное свойство используется, если активность **DatumNode** является вложенной в активности **TransactionScope**. Транзакция может быть следующих типов:
  - **Distributed** — распределенная транзакция, подразумевающая использование более одной транзакционной системы.
  - **Local** — последовательная транзакция, использующая одну транзакционную систему.
  - **None** — нет транзакции.
- **Command** — для указания команды, которая будет выполняться. Выбор команды осуществляется с помощью окна **Command** (см. рисунок 22). Формат команды следующий: <Решение> .<Проект> .<Сущность> .<Команда>.

**Примечание.** Объекты **Решения**, указанные в окне **Command** (см. рисунок 22), определены в **DatumNode**.

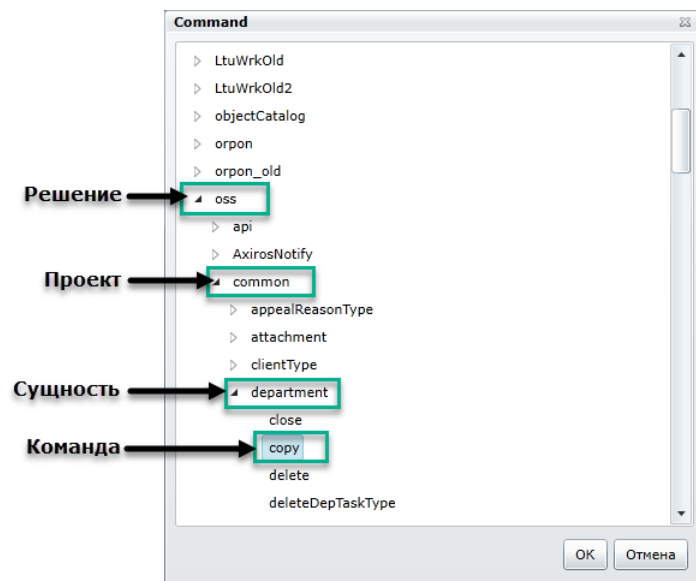



Рисунок 22 — Выбор команды для активности «DatumNode»

- **Connection** — для установки строки подключения **DatumNode** (по умолчанию используется значение \* . \*).

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 34 из 45

**Примечание.** 1. Формат строки следующий: <Ссылка на источник данных>. <Соединение>.  
 2. Значение поля **Connection** объекта **Соединение** переопределяет значение поля **Default Connection** объекта **Источник данных**.

- **MaxRetries** — для указания количества попыток вызова команды до получения положительного результата. Значение по умолчанию — 0 (одна попытка).
- **RetryDelay** — для указания времени ожидания между попытками выполнения команды.
- **Timeout** — для указания времени ожидания отклика от сервера при выполнении команды. По истечению этого интервала попытка выполнения команды считается неудачной.
- **Error** — содержит сообщение об ошибке, в случае если в ходе выполнения команды произошла ошибка.
- **IsCanceled** — содержит значение `true`, если произошла ошибка; или значение `false`, если ошибки не было.
- **Result** — содержит результат выполнения команды.
- **Table** — содержит результат выполнения команды в виде табличных данных, если был установлен тип операции **ExecuteReader**.
- **XML** — содержит результат выполнения команды в формате XML, если был установлен тип операции **ExecuteQuery**.

**Примечание.** Значение, содержащееся в свойстве **XML**, может быть десериализовано (`Context.Deserialize(of CustomType) (xml)`) в тип данных, определенный пользователем.

- **Parameters** — содержит список параметров команды.  
 Для каждого параметра можно задать следующие свойства:
  - **Ключ** — название параметра.
  - **Direction** — тип направления параметра:
    - **In** — входной параметр,
    - **Out** — выходной параметр,
    - **In/Out** — входной и выходной параметр.
  - **ArgumentType** — тип параметра.
  - **Expression** — выражение или название переменной рабочего процесса. Если установлена переменная рабочего процесса, то изменение переменной приводит к изменениям параметра.

## A.7.2 Активность «Task»


Активность **Task** (см. рисунок 23) предназначена для определения заданий, а также для возможности выделения фаз выполнения рабочих процессов.

Для данной активности доступны следующие свойства:

- **Наименование** — для указания произвольного названия активности.
- **AutoClose** — для включения (выключения) режима, при котором автоматически устанавливается статус активности в состояние **Выполнен**.

При установленном флажке рабочий процесс не останавливает выполнение, не выгружается из памяти и продолжает выполнение вложенных активностей.

- **HandleFault** — для включения (выключения) режима, при котором задание не будет отменено, даже если при выполнении задания возникла ошибка.
- **HandleUserFault** — для включения (выключения) режима, при котором перехватываются ошибки пользователей, но при этом задания не отменяются.
- **PersistBeforeSend** — для включения (выключения) режима, при котором определяется необходимость сохранения состояния экземпляра рабочего процесса перед выполнением задания.
- **Pulse** — для включения (выключения) режима, при котором активность становится системной.  
Активность **Task** с установленным флажком **Pulse** сигнализирует об окончании выполнения фазы экземпляра рабочего процесса.
- **Date** — для указания даты, которая будет сравниваться с датой обработки документа, породившего рабочий процесс. Если дата обработки меньше даты, значение которой соответствует `Date + Timeout`, то статус активности **Task** перейдет в состояние **Отменен** и активность не будет выполнена.
- **ExpectedDate** — для указания запланированной даты завершения задания (т. е. контрольный срок).
- **Name** — для указания типа задания (с помощью справочника **Типы заданий**).
- **Note** — для указания справочной информации об активности.
- **Period** — для указания временного интервала принудительной загрузки выгруженного рабочего процесса.
- **PublicCode, PublicId, System** — для указания внешних идентификаторов задания, которые требуются для передачи во внешнюю систему или выполнения задания по запросу из внешней системы.
- **Queue** — для указания названия очереди, в рамках которой будет выполнено задание.
- **Resource** — для указания ресурса во внешней системе.
- **ResourceType** — для указания типа внешнего ресурса.
- **TargetName** — не используется в текущей версии.
- **TargetPublicCode** — не используется в текущей версии.
- **TargetPublicId** — не используется в текущей версии.
- **TargetSystem** — не используется в текущей версии.
- **Timeout** — для указания времени ожидания выполнения активности.  
По истечении заданного интервала активность считается невыполненной, ее статус перейдет в состояние **Отменен**, при этом рабочий процесс продолжает выполнение.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 36 из 45

- **IsTransferred** — свойство, обозначающее, что сервис (содержащий данную активность) выбран для передачи управления в окне **Выполнение задания <номер>**.
- **Result** — содержит результат выполнения активности.
- **Parameters** — содержит список параметров активности.  
 Для каждого параметра можно задать следующие свойства:
  - **Ключ** — название параметра.
  - **Direction** — тип направления параметра:
    - **In** — входной параметр,
    - **Out** — выходной параметр,
    - **In/Out** — входной и выходной параметр.
  - **ArgumentType** — тип параметра.
  - **Expression** — выражение или название переменной рабочего процесса. Если установлена переменная рабочего процесса, то изменение переменной приводит к изменениям параметра.

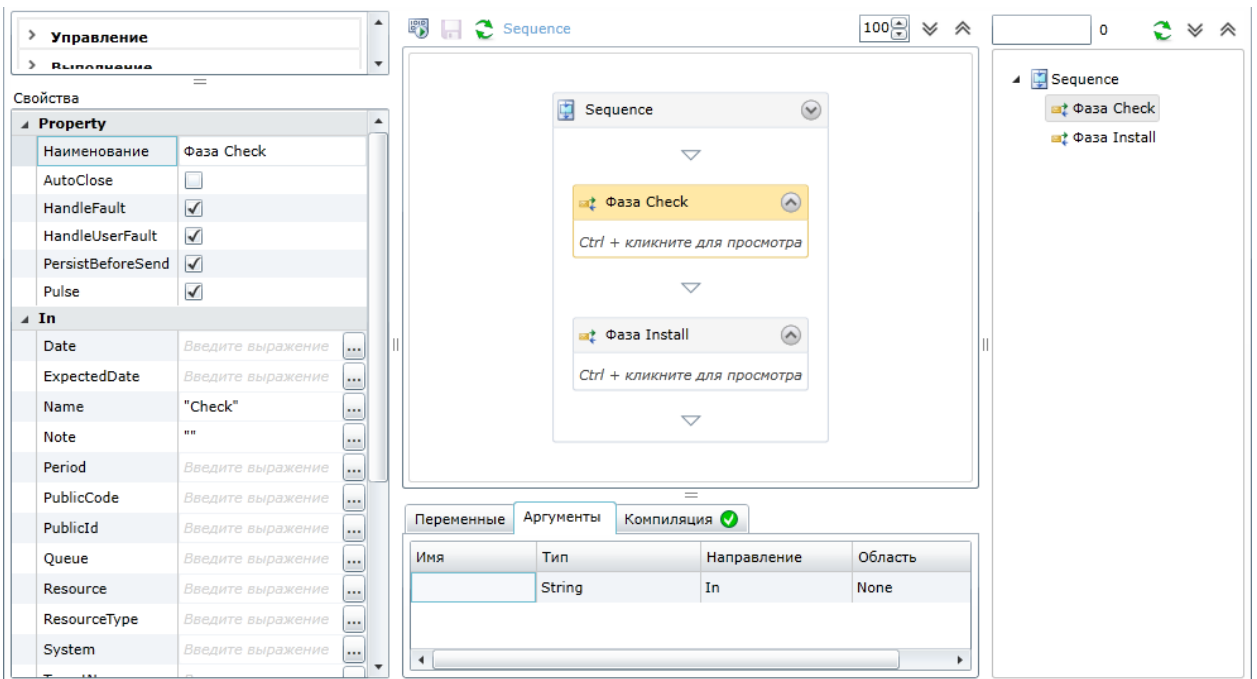


Рисунок 23 — Пример применения активности «Task»

## Приложение Б. Описание пользовательского интерфейса

Разработчик рабочих процессов системы «ТЕХНОГРАД КТП ИЗ» может использовать в своей работе все функции, реализуемые системой, но основная работа производится на вкладке **Процессы** (см. рисунок 24).

Для возможности разработки и отладки рабочих процессов учетная запись должна обладать ролью Разработчика OSS.

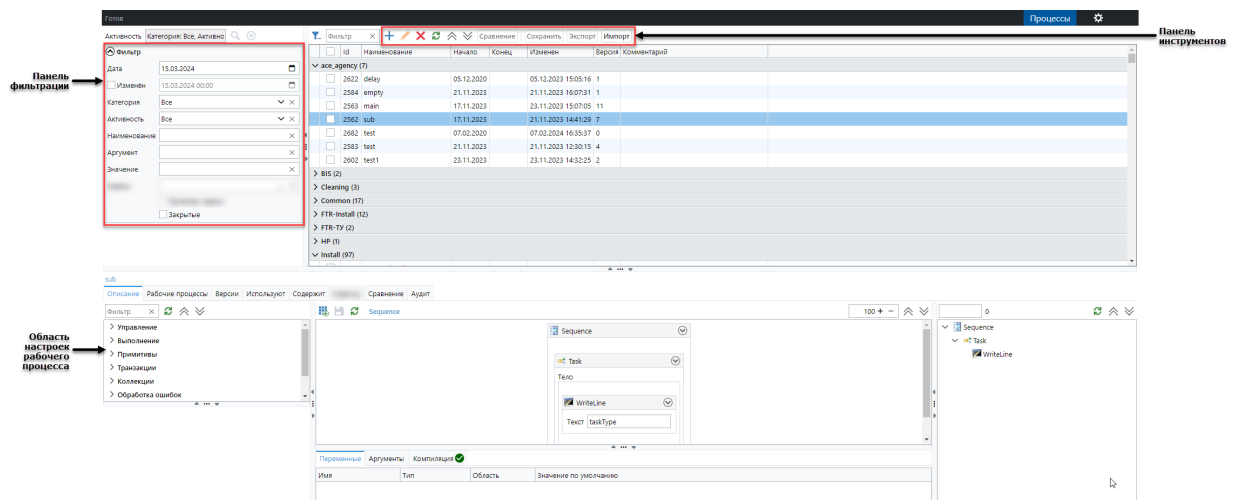


Рисунок 24 — Вкладка «Процессы»

На вкладке **Процессы** расположены следующие элементы:


- Панель фильтрации (см. Б.1).
- Панель инструментов (см. п. Б.2).
- Таблица процессов.
- Область настройки рабочего процесса (см. п. Б.3).

### Б.1 Панель фильтрации

Панель фильтрации предназначена для настройки условий отображения рабочих процессов в таблице. Данная панель содержит следующие элементы:

- Поле **Дата** — для отображения процессов, дата начала действия которых совпадает или меньше, указанного значения (по умолчанию используется текущая дата).
- Флажок **Изменен** — для включения (выключения) режима отображения процессов, которые были изменены до указанной даты (по умолчанию используется текущая дата).
- Выпадающий список **Категория** — для отображения процессов, относящихся к определенной категории.

**Примечание.** Названия категорий рабочих процессов задаются в соответствующем справочнике.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 38 из 45

- Выпадающий список **Активность** — для отображения процессов, которые содержат указанные активности.
- Поле **Наименование** — для отображения процессов, название которых совпадает с указанным значением (в том числе с учетом регистра).

**Примечание.** Для поиска процесса по неполному названию используется символ «%», который указывается до и (или) после названия.

- Поле **Аргумент** — для отображения процессов с учетом указанного названия свойства активности.
- Поле **Значение** — для отображения процессов с учетом указанного значения аргумента.

**Примечание.** Если в поле **Аргумент** не указано название свойства, то в таблице процессов будут отображены процессы, у которых указанные в поле **Значение** данные встречаются в любом месте процесса.

- Флажок **Закрытые** — для включения (выключения) режима отображения процессов, дата окончания которых указана и меньше даты, выбранной в поле **Дата**.

## Б.2 Панель инструментов

Панель инструментов содержит следующие кнопки:

- **Сравнение** — для сравнения свойств двух выбранных процессов. При этом процессы отображаются в формате XML на вкладке **Сравнение** области настройки рабочего процесса (см. п. Б.3.6).
- **Сохранить** — для сохранения выделенных рабочих процессов в файл формата XML, который может быть использован (при необходимости) в любом другом процессе.
- **Экспорт** — для сохранения выделенных рабочих процессов в файл.
- **Импорт** — для загрузки рабочего процесса из файла.


Кнопка **Сравнить** активна только в случае, если выбрано два процесса в таблице. Кнопки **Сохранить** и **Экспорт** активны, если выбран хотя бы один процесс в таблице.

## Б.3 Область настройки рабочего процесса

При выборе рабочего процесса в таблице в нижней части окна отображается область для его настройки.

Настройка рабочего процесса осуществляется на следующих вкладках:

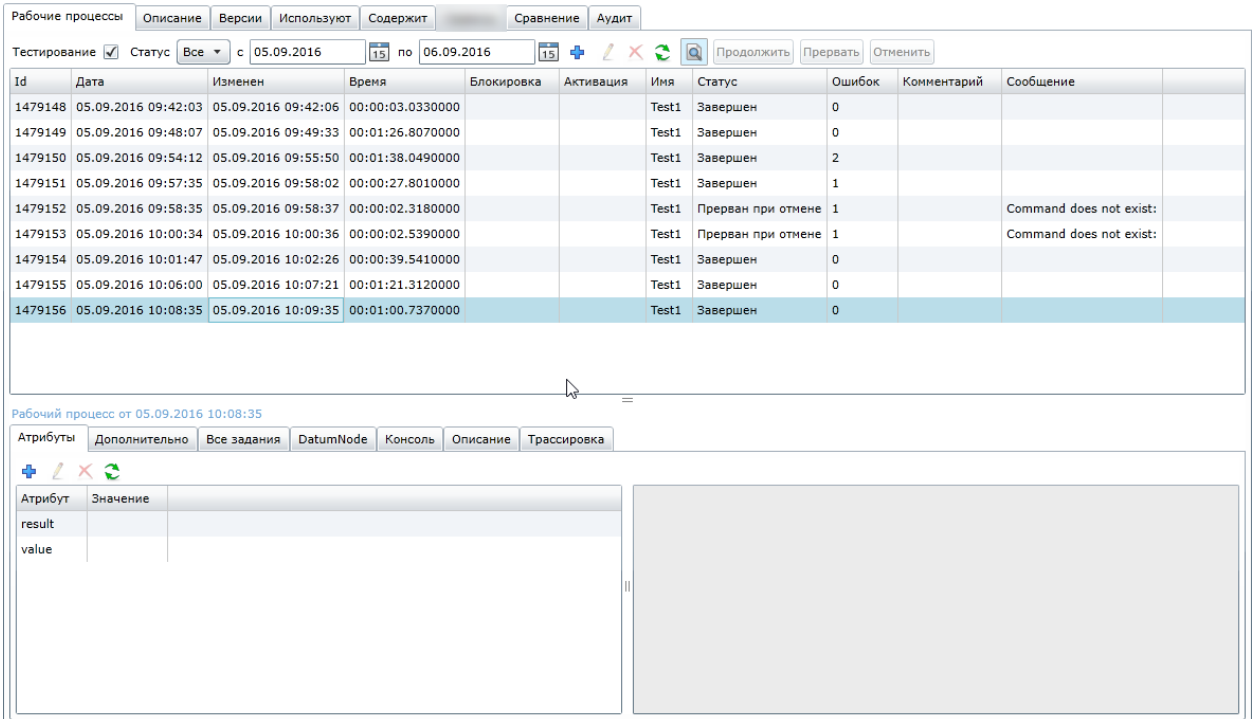
- Вкладка **Рабочие процессы** (см. п. Б.3.1).
- Вкладка **Описание** (см. п. Б.3.2)
- Вкладка **Версии** (см. п. Б.3.3).
- Вкладка **Используют** (см. п. Б.3.4).
- Вкладка **Содержит** (см. п. Б.3.5).
- Вкладка **Сравнение** (см. п. Б.3.6).

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
	Редакция: 1.0	2024 год

— Вкладка **Аудит** (см. п. Б.3.7).


### Б.3.1 Вкладка «Рабочие процессы»

Вкладка **Рабочие процессы** (см. рисунок 25) предназначена для настройки и проверки корректности выполнения процессов.




Id	Дата	Изменен	Время	Блокировка	Активация	Имя	Статус	Ошибок	Комментарий	Сообщение
1479148	05.09.2016 09:42:03	05.09.2016 09:42:06	00:00:03.0330000			Test1	Завершен	0		
1479149	05.09.2016 09:48:07	05.09.2016 09:49:33	00:01:26.8070000			Test1	Завершен	0		
1479150	05.09.2016 09:54:12	05.09.2016 09:55:50	00:01:38.0490000			Test1	Завершен	2		
1479151	05.09.2016 09:57:35	05.09.2016 09:58:02	00:00:27.8010000			Test1	Завершен	1		
1479152	05.09.2016 09:58:35	05.09.2016 09:58:37	00:00:02.3180000			Test1	Прерван при отмене	1		Command does not exist:
1479153	05.09.2016 10:00:34	05.09.2016 10:00:36	00:00:02.5390000			Test1	Прерван при отмене	1		Command does not exist:
1479154	05.09.2016 10:01:47	05.09.2016 10:02:26	00:00:39.5410000			Test1	Завершен	0		
1479155	05.09.2016 10:06:00	05.09.2016 10:07:21	00:01:21.3120000			Test1	Завершен	0		
1479156	05.09.2016 10:08:35	05.09.2016 10:09:35	00:01:00.7370000			Test1	Завершен	0		

Рисунок 25 — Вкладка «Процессы». Область настройки рабочего процесса. Вкладка «Рабочие процессы»

Для запуска выполнения рабочего процесса используется кнопка .

Управление выполнением рабочего процесса осуществляется с помощью следующих кнопок:


- **Продолжить** — для запуска прерванного или отмененного процесса.
- **Прервать** — для прерывания выполнения процесса.
- **Отменить** — для отмены выполнения процесса.


Для проверки свойств рабочего процесса используется кнопка , при этом в нижней части окна отображается группа вкладок, с помощью элементов которых возможно отследить возникшие ошибки и внести изменения в рабочий процесс.

Информация о конкретном тестируемом процессе (см. рисунок 25) содержится на следующих вкладках:

- **Атрибуты** — содержит информацию о заданных атрибутах процесса. При необходимости, могут быть добавлены новые атрибуты и (или) изменены значения существующих.
- **Дополнительно** — содержит справочную информацию о процессе.



	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 40 из 45

- **Все задания** — содержит список заданий (их количество соответствует количеству активностей типа Task).  
 На данной вкладке предоставляется возможность управлять выполнением заданий. Для просмотра свойств конкретного задания следует выбрать задание и щелкнуть кнопку .
- **DatumNode** — содержит список запросов к внешней системе, если таковые использовались в процессе.
- **Консоль** — содержит информацию об ошибках, возникших в ходе выполнения процесса, а также все сообщения, которые должны быть выведены на экран согласно процессу.
- **Описание** — содержит информацию об используемых в процессе активностях.
- **Трассировка** — позволяет проверить пошагово действия, который выполняет рабочий процесс (см. п. Б.3.1.1).




### *Б.3.1.1 Трассировка рабочего процесса*

При необходимости выяснения причины возникновения ошибки при выполнении рабочего процесса используется режим трассировки. При этом режиме доступна информация о каждом шаге выполнения.

Режим трассировки может быть использован для проверки выполнения всего процесса, а также любой активности в отдельности.

Действия, выполняемые процессом, отображаются на вкладке **Трассировка** снизу вверх (см. рисунок 26). Справа отображается окно описания активностей процесса со значками, обозначающими их статус.

Значки, обозначающие статус активностей, могут быть следующих видов:

-  — для обозначения активности, выполнение которой еще не окончено.
-  — для обозначения активности, в ходе выполнения которой не возникло ошибок.
-  — для обозначения активности, где произошла ошибка.

В нижней части вкладки **Трассировка** отображается информация, содержащая подробности выполнения каждого шага: описание выбранного шага в формате XML, список используемых переменных и аргументов.



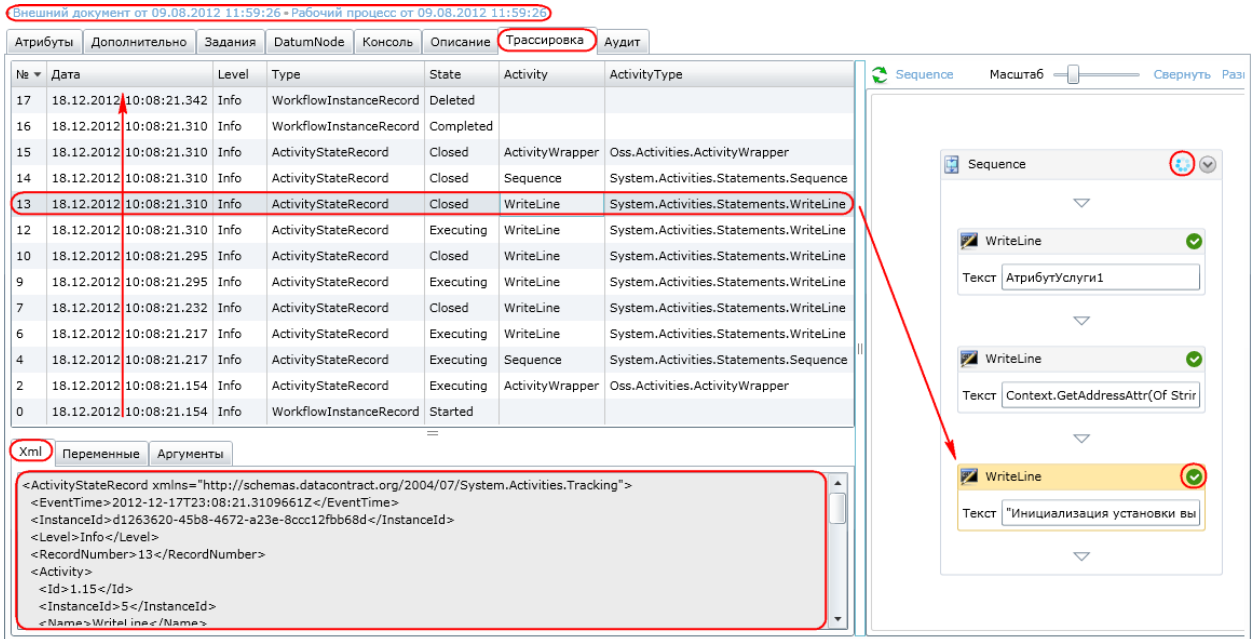


Рисунок 26 — Трассировка рабочего процесса

### Б.3.2 Вкладка «Описание»

Вкладка **Описание** (см. рисунок 27) предназначена для создания рабочего процесса в конструкторе с использованием активностей и других процессов.

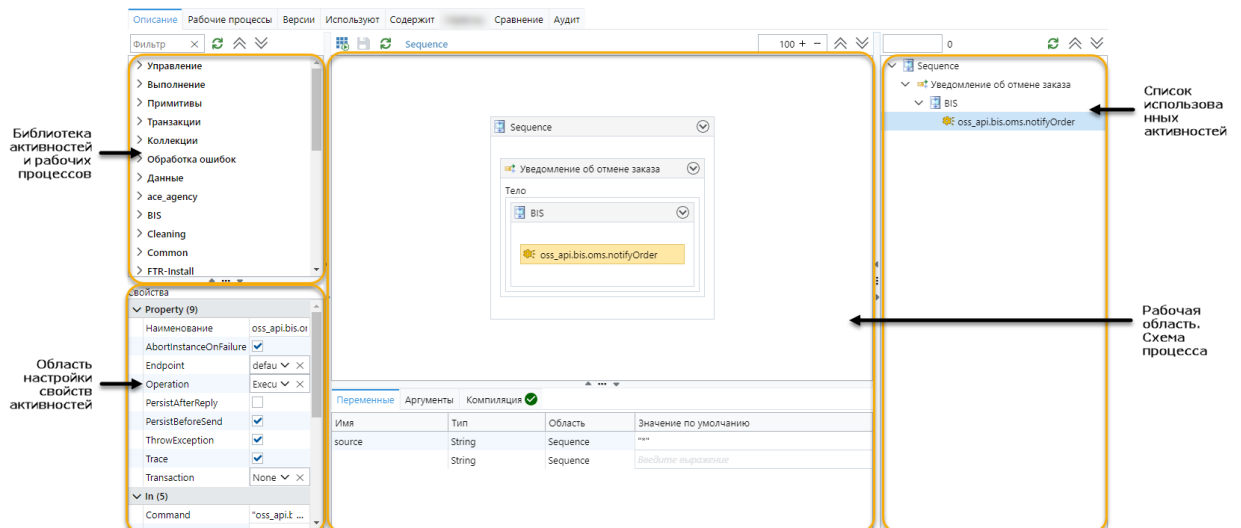




Рисунок 27 — Вкладка «Процессы». Область настройки рабочего процесса. Вкладка «Описание»

Создание рабочего процесса осуществляется путем перетаскивания необходимых активностей из библиотеки в рабочую область и установки свойств этих активностей. Для сохранения нового процесса и (или) внесенных изменений используется кнопка .

**Примечание.** Описание активностей и их свойств приведено в *Приложении А*.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 42 из 45

Для работы с активностями, расположенными в рабочей области, также используется контекстное меню, содержащее стандартные команды (**Открыть**, **Свернуть**, **Развернуть**, **Скопировать**, **Вставить**, **Вырезать**, **Удалить**), а также команду сохранения содержимого активности в файл формата XML.

В рабочей области расположены следующие вкладки:

- **Переменные** — для задания переменных, которые используются в рабочем процессе.

Для каждой переменной могут быть заданы следующие свойства:




- имя,
- тип,
- область видимости (название активности, которая является контейнером для других активностей),
- значение по умолчанию.


- **Аргументы** — для объявления аргументов, которые используются для передачи и получения данных рабочему процессу.

Для каждого аргумента могут быть заданы следующие свойства:

- имя,
- тип,
- тип направления:
  - **In** — входной аргумент,
  - **Out** — выходной аргумент,
  - **In/Out** — входной и выходной аргумент,
  - **Property** — значение аргумента всегда входное и является константой,
- область видимости:
  - **None** — аргумент доступен только внутри настраиваемого рабочего процесса,
  - **Local** — аргумент доступен внутри настраиваемого рабочего процесса, доступен для чтения (изменения) при помощи графического интерфейса или API,
  - **Document** — аргумент доступен внутри всех дочерних рабочих процессов документа для чтения (изменения) при помощи графического интерфейса или API,
  - **Context** — аргумент доступен для всех дочерних рабочих процессов группы документов и доступен для чтения (изменения) при помощи графического интерфейса или API.

**Примечание.** Описание свойств аргумента **Context** приведено в п.3.3.

- **Компиляция** — для отображения информации о ходе процесса компиляции. Компиляция настроенного рабочего процесса осуществляется с помощью кнопки . При успешной компиляции процесса на вкладке **Компиляция** отображается значок , в противном случае —  с указанием ошибок.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 43 из 45

### Б.3.3 Вкладка «Версии»

Вкладка **Версии** (см. рисунок 28) используется для просмотра всех версий рабочего процесса и при необходимости возврата к требуемой его версии.

Для управления версиями рабочего процесса используются следующие кнопки:

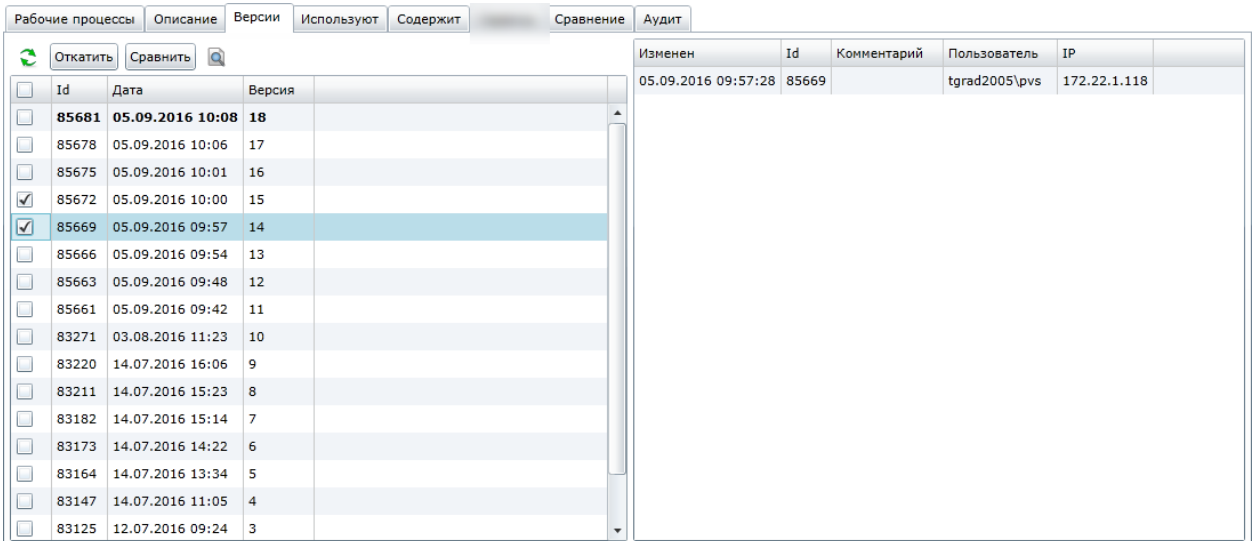
- **Откатить** — для возврата к выбранной версии процесса.
- **Сравнить** — для сравнения свойств двух выбранных процессов. При этом процессы отображаются в формате XML на вкладке **Сравнение** (см. пункт *Вкладка «Сравнение»*).

**Примечание.** Кнопка **Сравнить** активна только в случае, если выбрано две версии процесса.

В правой части окна отображается информация о выбранной версии процесса: дата внесения изменений в процесс, а также учетная запись, от имени которой были произведены изменения.

Для выделения отдельных версий используются следующие способы:

- Выделение полужирным начертанием — для обозначения текущей версии процесса.
- Выделение черным цветом — для обозначения ранних версий процесса.



Рабочие процессы			Описание	Версии	Используют	Содержит	Сравнение	Аудит										
<input type="button" value="Откатить"/> <input type="button" value="Сравнить"/>								<table border="1"> <thead> <tr> <th>Изменен</th> <th>Id</th> <th>Комментарий</th> <th>Пользователь</th> <th>IP</th> </tr> </thead> <tbody> <tr> <td>05.09.2016 09:57:28</td> <td>85669</td> <td></td> <td>tgrad2005\pvs</td> <td>172.22.1.118</td> </tr> </tbody> </table>	Изменен	Id	Комментарий	Пользователь	IP	05.09.2016 09:57:28	85669		tgrad2005\pvs	172.22.1.118
Изменен	Id	Комментарий	Пользователь	IP														
05.09.2016 09:57:28	85669		tgrad2005\pvs	172.22.1.118														
Id	Дата	Версия																
<input type="checkbox"/>	85681	05.09.2016 10:08	18															
<input type="checkbox"/>	85678	05.09.2016 10:06	17															
<input type="checkbox"/>	85675	05.09.2016 10:01	16															
<input checked="" type="checkbox"/>	85672	05.09.2016 10:00	15															
<input checked="" type="checkbox"/>	85669	05.09.2016 09:57	14															
<input type="checkbox"/>	85666	05.09.2016 09:54	13															
<input type="checkbox"/>	85663	05.09.2016 09:48	12															
<input type="checkbox"/>	85661	05.09.2016 09:42	11															
<input type="checkbox"/>	83271	03.08.2016 11:23	10															
<input type="checkbox"/>	83220	14.07.2016 16:06	9															
<input type="checkbox"/>	83211	14.07.2016 15:23	8															
<input type="checkbox"/>	83182	14.07.2016 15:14	7															
<input type="checkbox"/>	83173	14.07.2016 14:22	6															
<input type="checkbox"/>	83164	14.07.2016 13:34	5															
<input type="checkbox"/>	83147	14.07.2016 11:05	4															
<input type="checkbox"/>	83125	12.07.2016 09:24	3															


Рисунок 28 — Вкладка «Процессы». Область настройки рабочего процесса. Вкладка «Версии»

### Б.3.4 Вкладка «Используют»

Вкладка **Используют** предназначена для просмотра информации о процессах, которые используют (для своего выполнения) настраиваемый рабочий процесс.

### Б.3.5 Вкладка «Содержит»

Вкладка **Содержит** (см. рисунок 29) предназначена для просмотра информации о процессах, которые содержатся в настраиваемом рабочем процессе.

	ТЕХНОГРАД. КООРДИНАТОР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ. ИСПОЛНЕНИЕ ЗАКАЗА (ТЕХНОГРАД КТП ИЗ). Руководство разработчика рабочих процессов	
Редакция: 1.0	2024 год	Стр. 44 из 45

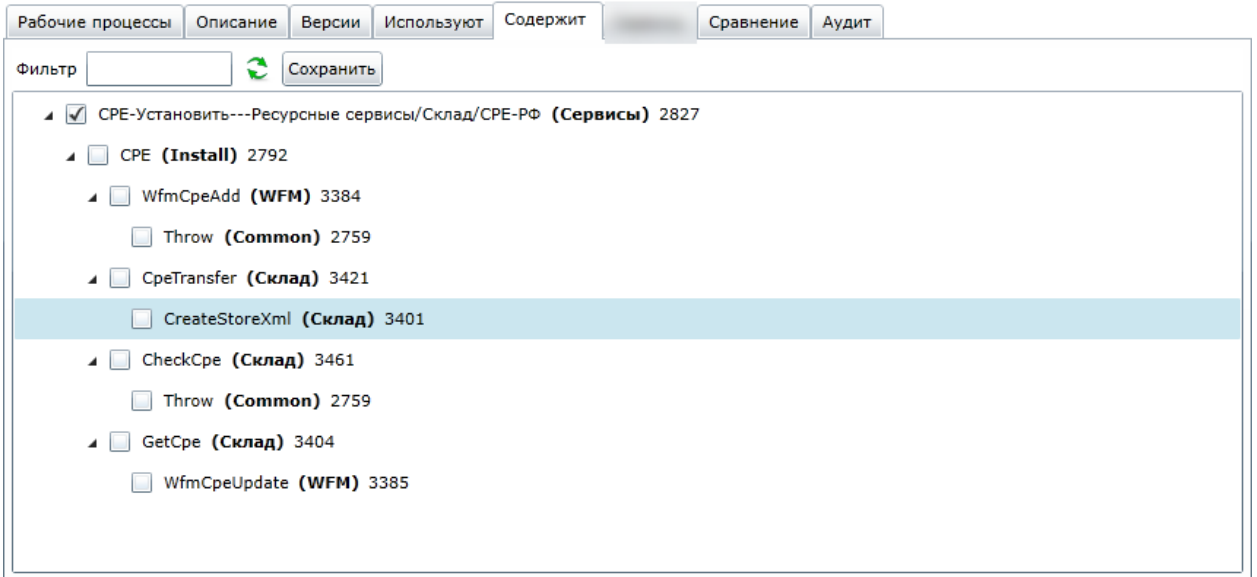


Рисунок 29 — Вкладка «Процессы». Область настройки рабочего процесса. Вкладка «Содержит»

### Б.3.6 Вкладка «Сравнение»

Вкладка **Сравнение** (см. рисунок 30) предназначена для сравнения двух разных процессов или двух разных версий одного процесса. Процессы отображаются в формате XML.

Для сравнения двух процессов (или двух версий процесса) следует выполнить следующие действия:

1. Выбрать в выпадающих списках **Процесс 1** и **Процесс 2** названия рабочих процессов.
2. Выбрать в выпадающих списках **Версия** номера версий сравниваемых процессов.
3. Щелкнуть кнопку **Сравнить**.

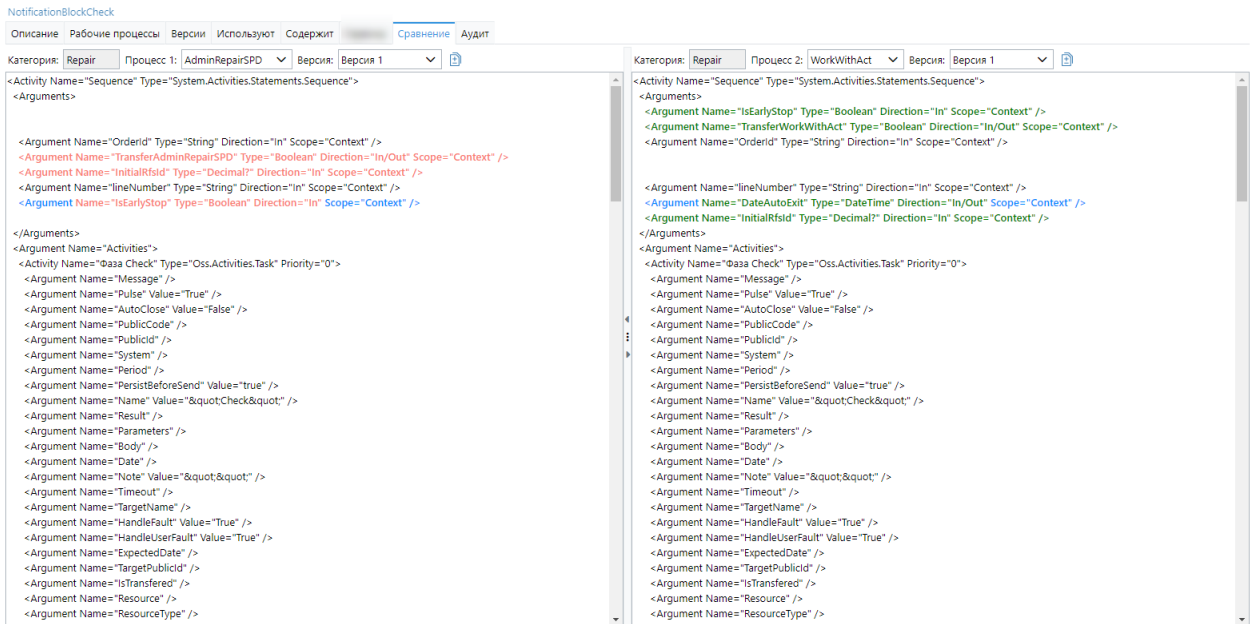


Рисунок 30 — Вкладка «Процессы». Область настройки рабочего процесса. Вкладка «Сравнение»

### Б.3.7 Вкладка «Аудит»

Вкладка **Аудит** предназначена для просмотра следующей информации:

- даты внесения изменений в процесс,
- идентификатора процесса,
- идентификатора версии процесса и дата ее создания,
- учетной записи, от имени которой были произведены изменения.