

ООО «АтомиСофт»

РУКОВОДСТВО НАСТРОЙЩИКА ПО РАБОТЕ С  
ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ  
«ПРИМА ИДК»

## СОДЕРЖАНИЕ

Условные обозначения и сокращения .....	3
1. Общие положения.....	4
3. Конфигурирование типов учетных единиц .....	11
4. Импорт и экспорт конфигурации.....	29
5. Конфигурирование операций .....	31
6. Конфигурирование журналов.....	39
7. Конфигурирование отчетов .....	41
8. Ведение справочников .....	44
9. Действия при аварийных ситуациях.....	46
Приложение 1 .....	48
Приложение 2 .....	50
Приложение 3 .....	77
Приложение 4.....	85
Приложение 5 .....	93
Приложение 6 .....	95
Приложение 7 .....	109
Приложение 8.....	112
Приложение 9 .....	115

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Сокращение (обозначение)	Расшифровка (пояснение)
АЭС	Атомная электростанция
ИДК	Индивидуальный дозиметрический контроль
ПО	Прима ИДК
СУБД	Система управления базами данных
УЕ	Учетная единица
УУП	Универсальная учетная платформа

# 1. ОБЩИЕ ПОЛОЖЕНИЯ

Руководство настройщика по работе с программным обеспечением «Прима ИДК» (далее – Руководство) содержит пошаговые инструкции и пояснения по основным операциям, выполняемым пользователем с ролью «Настройщик» в программном обеспечении «Прима ИДК» (далее – ПО). Основные возможности для пользователя с ролью «Настройщик» описаны в Разделе 2 настоящего документа.

## 1.1. Область применения

ПО применяется для обеспечения радиационной безопасности, ведения персонифицированного учета доз, контроля за соблюдением норм радиационной безопасности и основных санитарных правил.

ПО спроектировано как многопользовательское программное обеспечение на базе универсальной учетной платформы (далее – УУП) с соответствующей конфигурацией для ведения ИДК.

## 1.2. Основные возможности ПО

ПО предоставляет следующие основные возможности:

1) Учет и регистрация – обработка и долговременное хранение информации о персонале (прикрепление/открепление дозиметров), результатах измерений индивидуальных доз внешнего и внутреннего облучения, с обязательной фиксацией даты, времени и идентификатора средства измерения. Обеспечивается неизменность (невозможность удаления) однажды зарегистрированных данных.

2) Контроль – отслеживание фактических значений дозовых нагрузок сотрудников в сравнении с установленными контрольными уровнями. Автоматическая сигнализация о превышении допустимых уровней, а также контроль за своевременностью выдачи, возврата и списания (циклом обращения) дозиметров.

3) Мониторинг – предоставление информации в графическом и табличном виде о текущем радиационном состоянии персонала.

4) Отчетность – генерирование отчетных документов по формам, установленным национальным регулятором (например, форма №1-ДОЗ), а также форм внутренней отчетности для проведения анализа дозозатрат при планируемых работах и ведения индивидуальных карточек учета доз облучения персонала.

5) Планирование и нормирование – ведение базы данных с установленными дозовыми пределами и контрольными уровнями для различных категорий персонала и групп. Формирование данных о плановых дозиметрических обследованиях и поверок/калибровок средств ИДК, расчет прогнозируемых доз на планируемый период работ.

6) Взаимодействие с оборудованием – автоматический сбор, валидация и обработка данных, поступающих со средств автоматизированного ИДК:

- Термолюминесцентные дозиметры (ТЛД) и считывающие комплексы;
- Электронные прямопоказывающие дозиметры;
- СИЧ-системы;
- Турникеты и системы контроля доступа (СКУД) для прохода на основе актуальных доз.

7) Интеграция и обмен данными для автоматизации ручного ввода и обеспечения единого информационного пространства предприятия. ПО поддерживает механизмы обмена данными со следующими внешними системами и устройствами (но не ограничиваясь):

- Кадровые системы – импорт актуального списочного состава сотрудников, данных о перемещениях между подразделениями для автоматического обновления базы ИДК (интеграция через LDAP/OpenLDAP).
- Получение данных о радиационной обстановке в помещениях.
- Считыватели штрихкодов/QR-кодов / RFID-меток – для идентификации персонала и выданных дозиметров при выполнении операций выдачи/приема.
- Автоматизированные считывающие комплексы (ТЛД) – загрузка файлов с результатами измерений.

### 1.3. Уровень подготовки пользователя.

Пользователь обязан знать:

- настоящее Руководство;
- навыки работы с ОС;
- соответствующую терминологию настоящего документа;

### 1.4. Предварительные условия для работы с ПО.

Перед началом работы с ПО администратором установлено соответствующее программное обеспечение «Прима ИДК», входящее в поставку.

Убедиться, что на автоматизированном рабочем месте пользователя установлен браузер Google Chrome версии 105 и выше, или любой другой chromium-совместимый браузер (opera, яндекс браузер и т.д.), и пользователь имеет доступ к ПО в соответствии с его полномочиями.

В ПО зарегистрированы необходимые пользователи согласно «Руководству администратора по работе с программным обеспечением «Прима ИДК».



## 2. ОПИСАНИЕ ИНТЕРФЕЙСА

Главными задачами пользователя с ролью «Настройщик» является:

- описание типов учетных единиц в соответствии с технологическим процессом УЕ и другими особенностями учета УЕ в организации;
- конфигурирование операций, которые могут выполняться над учётными единицами;
- конфигурирование журналов, которые отображают изменение атрибутов учетных единиц;
- конфигурирование отчетов, которые возможно сформировать по данным учётных единиц;
- конфигурирование страниц мониторинга;
- ведение справочной информации.

Работа пользователей с ПО осуществляется через браузер, который установлен на локальном компьютере пользователя.

Для доступа к ПО пользователь вводит в адресную строку браузера соответствующий url, настроенный и выданный администратором.

После перехода по соответствующему адресу, пользователь попадает на страницу авторизации.

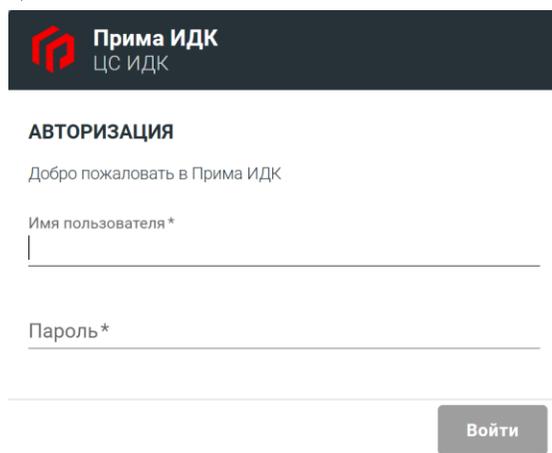


Рисунок 2.1. Страница авторизации пользователя

Вход пользователя осуществляется по личному логину и паролю, который выдается администратором. При первоначальном входе в ПО пользователю автоматически предлагается сменить пароль на свой личный.

При успешной аутентификации пользователя с ролью «Настройщик» в ПО открывается страница в разделе «Конфигуратор учетных единиц».

Интерфейс пользователя с ролью «Настройщик» состоит из следующих элементов:

- строка заголовка (верхняя часть) содержит название продукта и конфигурации, текущую дату и время, а также информационная панель пользователя и кнопку 

- рабочая область (центральная часть страницы), в которой отображается рабочая информация, которая меняется в зависимости от выбора в навигационном меню разделов и выполняемых операций.

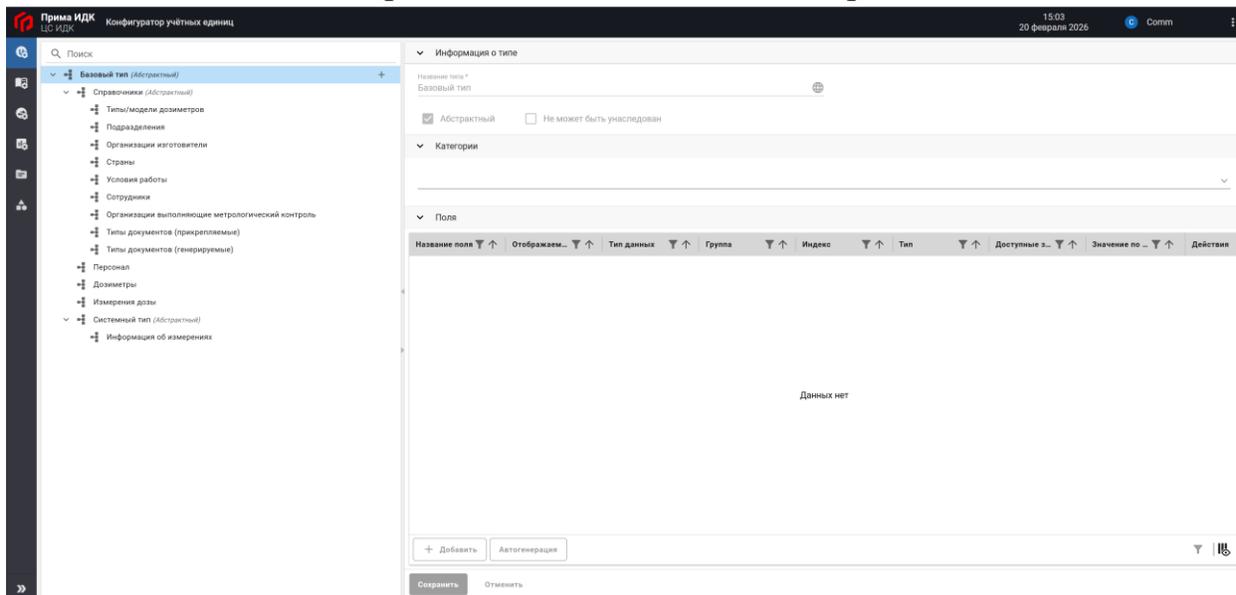


Рисунок 2.2. Пример интерфейса после авторизации пользователя с ролью «Настройщик»

Информационная панель пользователя по нажатию которой появляется функциональное подменю со следующим функционалом:

- отображение логина текущего пользователя;
- отображение сведений о сотруднике, привязанном к пользователю (в формате: Фамилия И.О.);
- переключение языка ПО;
- завершение текущей сессии работы с ПО кнопкой «Выход».

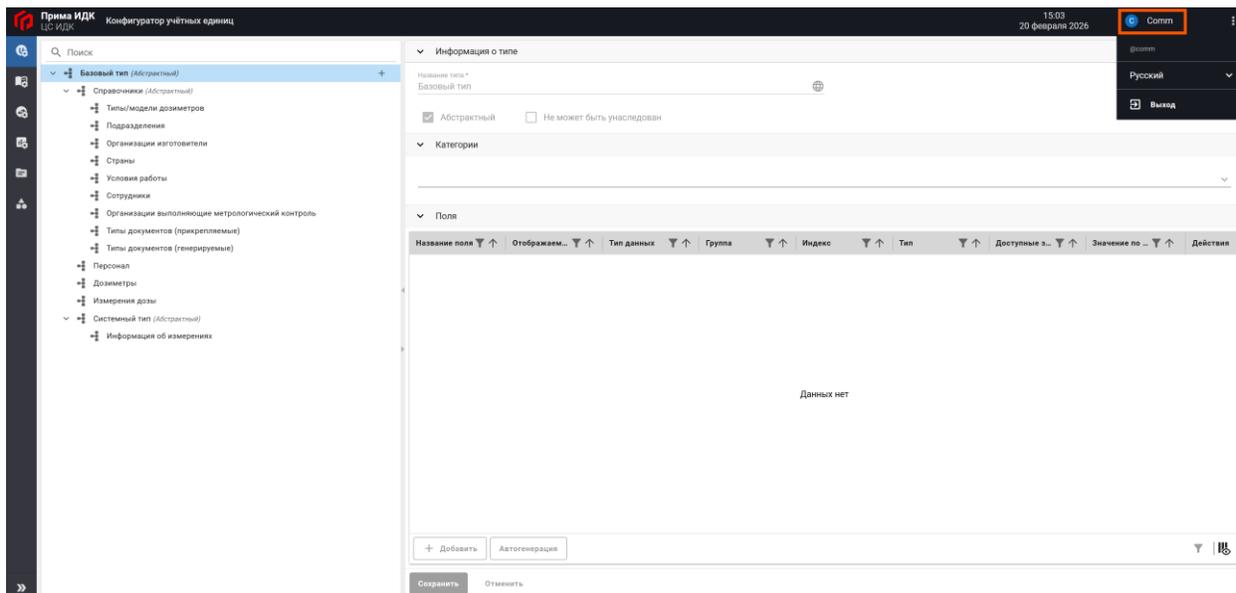


Рисунок 2.3. Пример подменю информационной панели

Кнопка , нажатие которой откроет выпадающий список с кнопками:

- Импорт;
- Экспорт;
- О программе;
- Помощь.

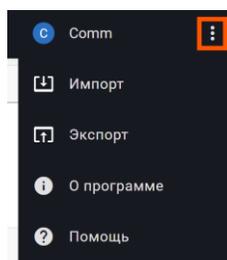
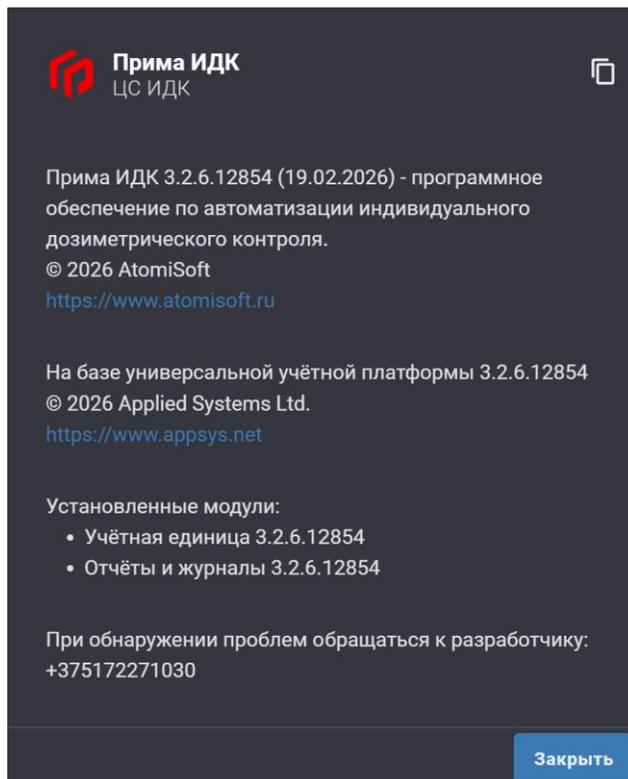


Рисунок 2.4. Подменю после нажатия кнопки 

Кнопки «Импорт» и «Экспорт» отвечают соответственно за импортирование из файла и экспортирование в файл конфигурации (Подробно описано в разделе 4).

Кнопка «О программе» ведет к диалоговому окну, которое содержит:

- версию ПО;
- дату выпуска;
- полное наименование ПО;
- информацию об изготовителе ПО;
- информацию об установленных модулях и их версия.



*Рисунок 2.5. Пример диалогового окна «О программе»*

Нажатие кнопки «Помощь» приводит к открытию страницы с Руководством в формате .pdf.

В левой нижней части страницы имеется кнопка открытия/скрытия главного навигационного меню (⏪), которое содержит следующие разделы:

- 1) Конфигуратор учетных единиц;
- 2) Конфигуратор операций;
- 3) Конфигуратор журналов;
- 4) Конфигуратор отчетов;
- 5) Справочники;
- 6) Категории.

### 3. КОНФИГУРИРОВАНИЕ ТИПОВ УЧЕТНЫХ ЕДИНИЦ

Описание конфигурации типов учетных единиц и их свойств выполняется в соответствующем специальном конфигураторе.

Интерфейс конфигуратора учетных единиц делится на две части:

- Левая часть содержит иерархическое дерево, с которыми будет работать пользователь с ролью «Настройщик».
- Правая часть конфигуратора содержит интерфейс, в котором указываются параметры и добавляются поля, описывающие свойства типов учетных единиц из дерева.

Дерево конфигурации представляет собой иерархическую модель данных, основанную на типах учетных единиц. Первоначально, конфигурация содержит один единственный «Базовый тип», от которого начинается ветвление. В данный тип невозможно вносить изменения, добавлять категории и поля.

Новый тип добавляется в необходимую ветвь дерева, если добавленный тип является дочерним, он наследует свойства и категории от родителя.

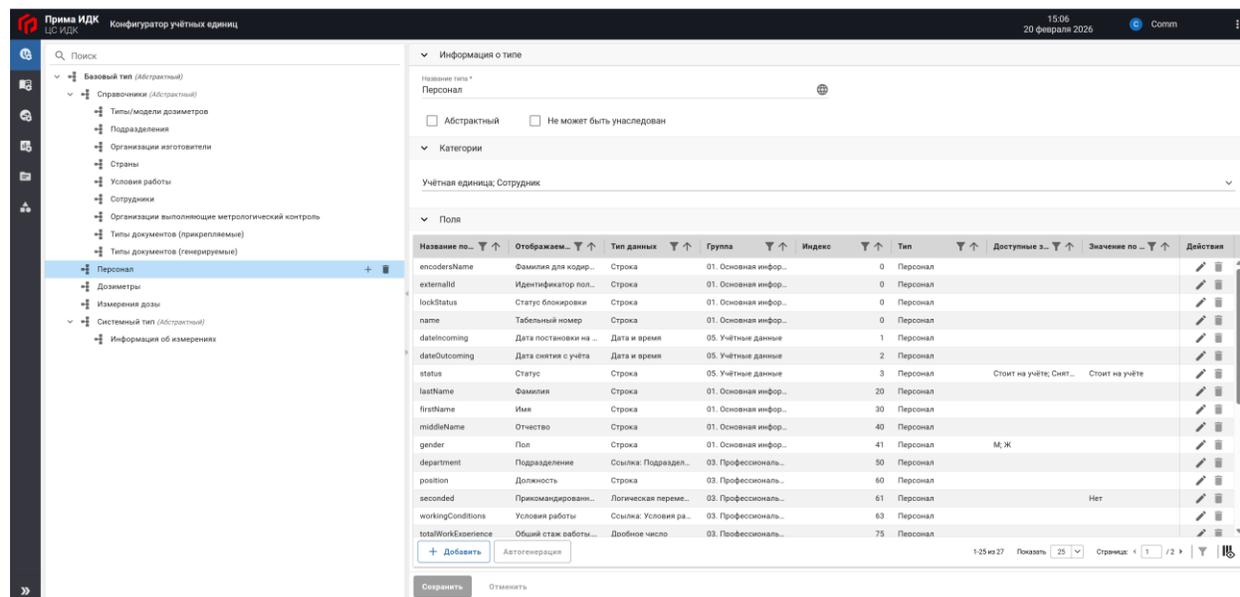


Рисунок 3.1. Пример интерфейса конфигурации по учету источников ионизирующего излучения

#### 3.1. Работа с типами

##### 1. Добавление нового типа происходит нажатием кнопки **Добавить**

(  ) на выделенном типе. После этого появляется новая строка. Далее необходимо ввести название типа в поле

«Название типа». После ввода нового названия необходимо нажать кнопку **Сохранить** для добавления нового типа в дерево конфигурации.

2. Отмена добавления нового типа происходит нажатием на кнопку **Отменить**.

3. Для удаления существующего типа следует выделить необходимый тип и нажать кнопку **Удалить** (  ).

4. Подтвердить удаление с помощью кнопки **Сохранить** (  ).

Любые изменения, которые производит пользователь над типом можно сохранить, путем нажатия кнопки **Сохранить**, или отменить, при нажатии кнопки **Отменить**.

Каждый тип имеет дополнительные параметры, которые представлены в таблице и настраиваются по усмотрению пользователя:

Наименование параметра	Описание
Абстрактный	Параметр указывающий, что данный тип не может быть объектом для взаимодействия.
Не может быть унаследован	Параметр указывающий, что для данного типа нельзя создавать наследника.

### 3.2. Работа с категориями

Категория – это требования, которые гарантируют наличие необходимых полей и их характеристик, которые должны быть у определённого типа. Также категория может активировать необходимую логику для типа учетной единицы.

Категории в ПО могут быть платформенными, либо «приносятся» модулями, которые обладают необходимой логикой для совершения действий в ПО.

Список доступных для работы категорий в ПО «Настройщик» может посмотреть на вкладке «Категории».

Название	Описание	Поля категории	Модуль
Справочник	Платформенная категория, которая указывает, что тип учётной единицы «...	name, status	Универсальная учётная платформа
Документы прикрепляемые	Учётные единицы, которые имеют данную категорию, могут быть указаны «...	name, group, fileType, setCurrentDate, isDocumentNumberRequired, includeAll...	Универсальная учётная платформа
Структура	Категория для создания специальных типов учётных единиц, которые могут...	structId, location	Универсальная учётная платформа
Сотрудник	Категория, назначенная справочнику, хранящему списки сотрудников, которые...	name, firstName, lastName, middleName, encodersName, externalId, position, s...	Универсальная учётная платформа
Права на доступ к учётным единицам	Категория для типа, учётные единицы которого являются областями досту...	name, fieldName	Универсальная учётная платформа
УИВ	Учётные единицы, которые имеют данную категорию, используются как ус...	name, tidIntegrity, tidCondition, reusable, tidInstallationDate, tidInstallationPerfo...	Универсальная учётная платформа
Объект применения УИВ	Категория для учётных единиц на которых применяется УИВ.	tidPlace, location	Универсальная учётная платформа
Учётная единица	Базовая учётная единица.	name, status, lockStatus, dateIncoming, dateOutcoming	Учётная единица
Учётная единица. Контейнер	Учётная единица, в которую могут помещаться другие учётные единицы.		Учётная единица
Учётная единица. Инспекция	Категория для создания типа, учётной единицей которого является проце...	dateIncoming, dateOutcoming, department, inspectionItems	Учётная единица
Учётная единица. Инвентаризация	Категория для создания типа, учётной единицей которого является проце...	dateIncoming, dateOutcoming, department, inventoryItems	Учётная единица
Учётная единица. Объект инвентаризации	Учётная единица, которая может быть проверена во время инвентаризаци...	inventories	Учётная единица
Учётная единица. Перегруппировка	Категория для типов учётных единиц, которые могут являться исходными «...	rebatchedParentItem, rebatchedChildItem	Учётная единица

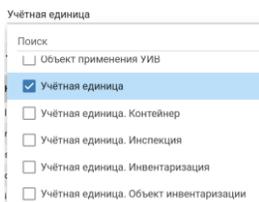
Рисунок 3.2. Пример активных (доступных) категорий на вкладке «Категории»

Для присваивания (активации) категории для типа учетной единицы необходимо:

1. Создать новый тип (см. раздел 3.1 настоящего Руководства), либо выбрать необходимый существующий тип учетной единицы.

2. Развернуть строку  Категории и нажать на кнопку  в открывшемся окне.

3. В выпадающем списке необходимо выбрать одно или несколько



значений

4. Для подтверждения действия необходимо нажать кнопку **Сохранить**.

Некоторые категории могут потребовать наличия у данного типа обязательных полей.

Их можно добавить вручную или воспользоваться кнопкой **Автогенерация**. После генерации необходимых полей, пользователю остаётся заполнить недостающие значения.

Более подробная информация про работу с полями описывающих тип учётной единицы, содержится в разделе 3.3 настоящего Руководства.

Несколько категорий могут требовать наличия поля с одним и тем же именем. Добавление одного поля с требуемым именем является необходимым и достаточным условием работы нескольких категорий.

Удаление (деактивация) категории с типа учетной единицы:

1. Выбрать нужный тип.
2. Развернуть список категорий.
3. В выпадающем списке снять галочки с ненужных категорий.
4. Для подтверждения действия необходимо нажать кнопку

**Сохранить.**

### 3.2.1. Категории платформы

Базовая функциональность ПО поддерживает следующие платформенные категории:

- Справочник – платформенная категория, которая необходима для присвоения типу учетной единицы статуса «Справочник» и отображение его на вкладке «Справочники» в конфигураторе.

Необходимые поля для категории «Справочник»:

Название поля	Описание
name	Наименование записи в справочнике.
status	Учётный статус справочника.

- Сотрудник – платформенная категория, которая формирует список сотрудников для привязки к учетной записи пользователя.

Необходимые поля для категории «Сотрудник»:

Название поля	Описание
name	Фамилия и инициалы сотрудника
firstName	Имя сотрудника
lastName	Фамилия сотрудника
middleName	Отчество сотрудника
encodersName	Фамилия и инициалы сотрудника на английском языке
position	Должность сотрудника

- Права на доступ к учётным единицам – платформенная категория, с помощью которой формируется список областей доступа для работы с учётными единицами. Область доступа назначается пользователю с ролью «Учетчик» для разделения полномочий. Каждая область доступа отображается в разделе «Группы прав» у администратора.

Необходимые поля для категории «Права на доступ к учётным единицам»:

Название поля	Описание
name	Наименование области доступа
firstName	Фильтрационное поле. В данном поле указывается название поля, по значениям которого необходимо фильтровать учётные единицы. Если поле, название которого указано в fieldName, отсутствует у учётной единицы, то по умолчанию такая учётная единица попадает в итоговую выборку значений. Поле скрыто по умолчанию в отображении справочников для Учётчика. Пример: значение fieldName = department, означает, что все учётные единицы, у которых существует поле department фильтруются по значению из поля name этой записи справочника.

- Документ прикрепляемый – платформенная категория, с помощью которой формируется список прикрепляемых типов документов на вкладке «Документы» при проведении операции.

Необходимые поля для категории «Документ прикрепляемый»:

Название поля	Описание
name	Наименование документа.
fileTypes	Типы файлов, которые можно прикрепить
group	Группа документов

- Документ генерируемый – платформенная категория, с помощью которой формируется список генерируемых Системой типов документов на вкладке «Документы» при проведении операции.

Необходимые поля для категории «Документ генерируемый»:

Название поля	Описание
name	Наименование документа.
template	Шаблон (прикрепляемый файл).

Конфигурирование шаблона excel выполняется согласно приложению 5

- «Структура» является платформенной категорией, которая необходима для отображения информации в учетной единице в виде таблицы.

Каждое поле в типе учетной единицы с активной категорией «Структура», будет являться наименованием графы в таблице. Количество граф будет зависеть от количества созданных полей.

Например: тип учетной единицы с активной категорией «Структура» содержит следующие поля: Радионуклид, Активность. Такой тип будет в интерфейсе пользователя с ролью «Учетчик» отображаться следующим в виде таблицы с графами: Радионуклид, Активность.

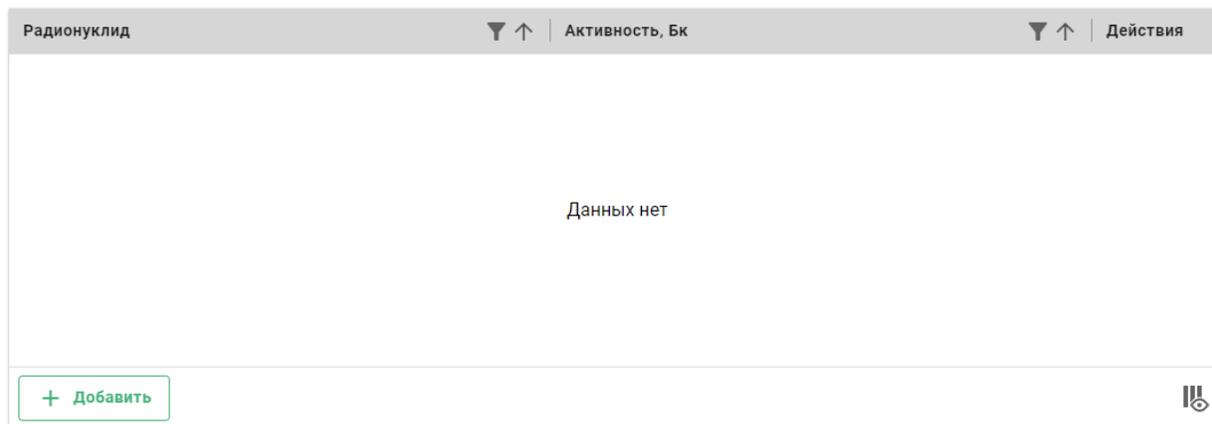


Рисунок 3.3. Пример отображения структуры у пользователя с ролью «Учетчик»

Необходимые поля для категории «Структура»:

Название поля	Описание
stuctId	Системное поле в виде счетчика, которое не заполняется и скрыто для пользователей. Данное поле необходимо для идентификации записей в таблице.

- «Объект применения УИВ» - платформенная категория, с помощью которой формируется список учётных единиц, на которые могут устанавливаться учётные единицы с категорией «УИВ».

Необходимые поля для категории «Объект применения УИВ»:

Название поля	Описание
tidPlace	Место размещения УИВ
location	Учётная единица, в которую помещена описываемая учётная единица. Например, контейнер или сейф.

- «УИВ» – платформенная категория, с помощью которой формируется список УИВ, которые можно установить на учётные единицы с категорией «Объект применения УИВ».

Необходимые поля для категории «УИВ»:

Название поля	Описание
name	Номер УИВ
tidIntegrity	Целостность УИВ
tidCondition	Состояние УИВ
applicationObject	УЕ, на которой применен УИВ
reusable	Многоразовый УИВ
tidInstallationDate	Дата и время установки УИВ
tidInstallationPerformer	Исполнитель установки УИВ
tidInstallationAttester	Свидетель установки УИВ
tidRemovalDate	Дата и время снятия УИВ
tidRemovalPerformer	Исполнитель снятия УИВ
tidRemovalAttester	Свидетель снятия УИВ

### 3.2.2. Категории модуля «Учетная единица»

- Учетная единица – основная категория модуля. Типы УЕ с категорией «Учётная единица» могут быть зарегистрированы в операции «Регистрация учётной единицы» и всех других операциях на ее базе, а также могут быть отредактированы в операции «Изменение свойств» и всех других операциях на ее базе.

Необходимые поля для категории «Учетная единица»:

Название поля	Описание
name	Название учётной единицы
status	Учётный статус УЕ. Принимает одно из двух значений: – Стоит на учете; – Снят с учета.
lockStatus	Статус УЕ, характеризующий доступность этой УЕ для операции.
dateIncoming	Дата постановки на учёт
dateOutcoming	Дата снятия с учета

- Учетная единица. Контейнер – в учётных единицах с категорией «Учётная единица. Контейнер» могут размещаться другие учётные единицы. Учётные единицы, расположенные в контейнере, снимаются с учёта вместе с контейнером. При этом, размещение может быть не обязательно одного уровня вложенности: если снять с учёта помещение, то снимется с учёта и шкаф в помещении, и сейф в шкафу, и УЕ в сейфе.

Необходимых полей для категории нет.

- Учётная единица. Инвентаризация – учётные единицы с данной категорией могут быть зарегистрированы в операции «Инвентаризация».

Необходимые поля для категории «Учётная единица. Инвентаризация»:

Название поля	Описание
dateIncoming	Дата начала инвентаризации
dateOutcoming	Дата окончания инвентаризации
inventoryItems	Объекты инвентаризации
department	Структурное(ые) подразделение(ия), в котором(ых) проходила инвентаризация

- «Учетная единица. Инспекция» – учётные единицы с данной категорией могут быть зарегистрированы в операции «Инспекция».

Необходимые поля для категории «Учётная единица. Инспекция»:

Название поля	Описание
dateIncoming	Дата начала инспекции
dateOutcoming	Дата окончания инспекции
inventoryItems	Объекты инспекции
department	Структурное(ые) подразделение(ия), в котором(ых) проходила инспекция

- Учетная единица. Объект инвентаризации – учётные единицы с категорией «Учётная единица. Объект инвентаризации» имеют дополнительную вкладку в своей карточке «Инвентаризация/Инспекция». На дополнительной вкладке отображается в каких инвентаризациях/инспекциях участвовала УЕ, карточку которой открыли для просмотра.

Необходимые поля для категории «Учётная единица. Объект инвентаризации»:

Название поля	Описание
---------------	----------

inventories	Все инвентаризации и инспекции, в которых участвовала данная учётная единица в числе inventoryItems или inspectionItems.
-------------	--

- Учетная единица. Перегруппировка – учётные единицы с категорией «Учётная единица. Перегруппировка» могут участвовать в операциях по перегруппировке учётных единиц или являться результатом этой операции.

- Необходимые поля для категории «Учётная единица. Перегруппировка»:

Название поля	Описание
rebatchedParentItem	Исходные (родительские) учётные единицы в перегруппировке.
rebatchedChildItem	Конечные (дочерние) учётные единицы в перегруппировке.

### 3.3. Работа с полями

Поля представляют собой набор информации, который описывает свойства отдельного типа.

Каждому типу, кроме «Базового типа», могут быть добавлены поля.

Для этого необходимо:

- Развернуть строку  Поля
- Нажать кнопку  Добавить в открывшемся окне.
- В открывшемся окне «Добавление нового поля» заполнить данные.

данные.

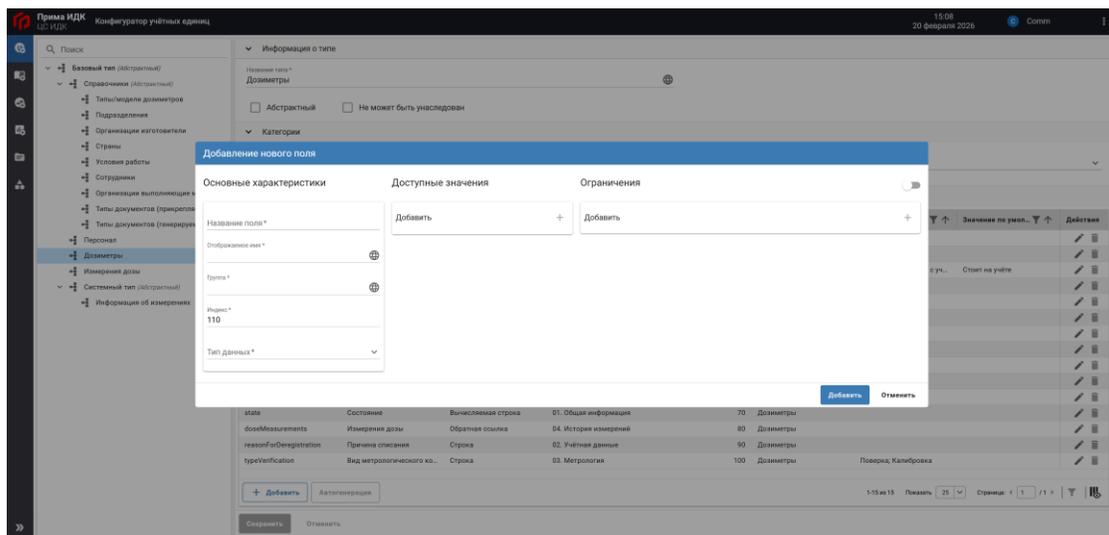


Рисунок 3.4. Интерфейс по добавлению нового поля

4. Нажать кнопку .

Для отмены создания нового поля необходимо нажать кнопку **Отменить**.

В графе таблицы «Действия» расположены элементы управления (кнопки):

-  – редактировать;
-  – удалить.

При нажатии на кнопку **Удалить** поле окрашивается в красный цвет и появляются следующие элементы подтверждения в графе «Действия»:

-  – Сохранить;
-  – Отменить.

Окно «Добавление нового поля» содержит основной набор параметров, который приведён в таблице:

Наименование параметра	Описание
Название поля	Системное название поля. Поле является обязательным для заполнения. Максимальное число символов — 64. Наименование поля должно быть уникальным. Разрешается в названии использовать только одно слово, состоящее из букв латинского алфавита, цифры.
Отображаемое имя	Наименование, которое будет отражаться в интерфейсе пользователя. Поле является обязательным для заполнения. Максимальное число символов — 256. Разрешены буквы латинского и кириллического алфавита, цифры и спец.символы.
Группа	Группа, к которой принадлежит поле. Поле является обязательным для заполнения. Максимальное число символов — 64. Разрешены буквы латинского и кириллического алфавита, цифры и спец.символы.
Тип данных	Тип данных, который выбирается из выпадающего списка значений. Возможные значения:

	<ul style="list-style-type: none"> <li>• Строка;</li> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Логическая переменная;</li> <li>• Ссылка;</li> <li>• Обратная ссылка;</li> <li>• Файл;</li> <li>• Таблица;</li> <li>• Текст;</li> <li>• Локализованная строка;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul> <p>Можно выбрать только одно значение.</p> <p>Описание поддерживаемых типов данных выполнено в п.2.4.</p>
<p>Множественные значения</p>	<p>Указывает количество значений, которые может принимать поле и может принимать два состояния: Да/Нет.</p> <p>Нет – поле состоит из одного значения. При активном параметре «Допустимые значения» из списка возможно выбрать только одно значение.</p> <p>Да – одно поле может содержать одно или более значений. При активном параметре «Допустимые значения» из списка можно выбрать одно или несколько значений.</p> <p>Параметр не отображается, если выбран тип данных:</p> <ul style="list-style-type: none"> <li>• Файл;</li> <li>• Обратная ссылка;</li> <li>• Логическая переменная;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>

Поле уникальное	<p>Указывает, что значение в поле не может повторяться и является уникальным среди данного типа учётной единицы и его дочерних типов (при их наличии), которые наследуют данное поле, если уникальность определена в родительском типе.</p> <p>Уникальность проверяется вне зависимости от статуса учетных единиц.</p> <p>При проверке на уникальность учётная единица не сравнивается сама с собой.</p> <p>Параметр не отображается, если выбран тип данных:</p> <ul style="list-style-type: none"> <li>• Логическая переменная;</li> <li>• Текст;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Ссылка;</li> <li>• Дата;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>
Поле обязательное	<p>Указывает, что поле должно быть обязательным для заполнения.</p> <p>Параметр не отображается, если выбран тип данных:</p> <ul style="list-style-type: none"> <li>• Логическая переменная;</li> <li>• Обратная ссылка;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>
Поле только для чтения	<p>Значения для данного поля можно внести однократно при проведении операции «Регистрация учётной единицы» или в операциях на ее основе.</p> <p>Далее, при работе с объектом, значение в данном поле будет использоваться только для чтения и не может быть изменено. Однако, пользователь может выделять его и копировать значение из него.</p> <p>Параметр не отображается, если выбран тип данных:</p>

	<ul style="list-style-type: none"> <li>• Таблица;</li> <li>• Обратная ссылка;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>
Поле заблокировано	<p>Указывает, что поле неактивно, пользователь не может взаимодействовать.</p> <p>Отсутствует у типов данных: Структура, Файл, Обратная ссылка.</p> <p>Тип данных, которые заблокированы всегда:</p> <ul style="list-style-type: none"> <li>• Обратная ссылка;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>
Автозаполняемое поле	<p>Параметр имеет значение «Да» или «Нет» и отображается, если выбран тип данных: Строка, Текст, Целое число, Дробное число, Дата и время, Дата.</p> <p>Указывает, что значение в поле может вычисляться Системой по заданной формуле valuePath (см. Приложение 1) на основе других полей.</p> <p>Список поддерживаемых функций и операторов указан в приложении 2. Описание работы автозаполняемого поля для пользователя с ролью «Учетчик» указано в приложении 7.</p>
Индекс	Указывает, какой приоритет отображения будет иметь добавляемое поле в пользовательском интерфейсе.
Скрыть поле в операциях и карточках	Поле, которое не отображается для заполнения и взаимодействия, а также отсутствует в карточке учётной единицы.

### 3.4. Типы данных

«Тип данных» является обязательным для заполнения параметром. Данное ПО использует типы данных, которые приведены ниже:

Тип данных	Описание
------------	----------

Строка	Тип, при выборе которого доступен ввод любых символов.
Локализованная строка	Тип, при выборе которого доступен ввод любых символов и используется для отображения строки в зависимости от выбранного языка системы. Данный тип данных используется при необходимости поддержки мультиязычности пользователей в ПО. Подробное описание работы локализованной строки представлено в приложении 7.
Текст	Тип, при выборе которого доступен ввод любых символов и используется для отображения многострочного текста.
Логическая переменная	Тип, при котором могут приниматься только два значения: «Да» или «Нет».
Целое число	Тип, при котором для ввода доступны только целочисленные значения.
Дробное число	Тип, при котором для ввода доступны числа дробной часть. При этом отображаются только два знака после запятой. При вводе большого значения происходит округление и отображаемое значение принимает вид: 4.00e+5, что равно 400000.
Дата	Тип данных, при котором для ввода доступны только дата.
Дата и время	Тип, при котором для ввода доступны только дата и время.
Ссылка	Тип данных, при котором происходит ссылка на неабстрактный тип учетной единицы из дерева конфигурации.
Обратная ссылка	В поле с таким типом данных отображаются учётные единицы, которые имеют ссылки на данную учётную единицу. Данные в таком поле всегда заблокированные и представлены в виде ссылки на учетные единицы.
Таблица	Тип, при котором происходит ссылка на неабстрактный тип с категорией «Структура».

	Если этот параметр присвоить учётной единице, то при работе с ней данное поле будет отображаться в виде таблицы
Файл	<p>Добавление в поле неисполняемых файлов размером не превышающим 50 МБ.</p> <p>Разрешённые форматы:</p> <ul style="list-style-type: none"> <li>• pdf;</li> <li>• docx;</li> <li>• xlxs;</li> <li>• тексты;</li> <li>• картинки.</li> </ul>
Вычисляемая строка	<p>Указывает, что значение в поле вычисляться Системой по заданной формуле (указанному valuePath) и приводит результат вычисления к типу данных «Строка»</p> <p>Расчет значения в поле происходит автоматически при открытии формы с участием такого поля и не хранится в БД.</p>
Вычисляемое целое число	<p>Указывает, что значение в поле вычисляться Системой по заданной формуле (указанному valuePath) и приводит результат вычисления к типу данных «Целое число»</p> <p>Расчет значения в поле происходит автоматически при открытии формы с участием такого поля и не хранится в БД.</p>
Вычисляемое дробное число	<p>Указывает, что значение в поле вычисляться Системой по заданной формуле (указанному valuePath) и приводит результат вычисления к типу данных «Дробное число»</p> <p>Расчет значения в поле происходит автоматически при открытии формы с участием такого поля и не хранится в БД.</p>
Вычисляемая дата и время	<p>Указывает, что значение в поле вычисляться Системой по заданной формуле (указанному valuePath) и приводит результат вычисления к типу данных «Дата и время»</p> <p>Расчет значения в поле происходит автоматически при открытии формы с участием такого поля и не хранится в БД.</p>

Вычисляемая дата	Указывает, что значение в поле вычисляться Системой по заданной формуле (указанному valuePath) и приводит результат вычисления к типу данных «Дата» Расчет значения в поле происходит автоматически при открытии формы с участием такого поля и не хранится в БД.
------------------	--

В зависимости от значения, которое присвоит пользователь параметру «Тип данных», могут быть добавлены дополнительные параметры:

Дополнительный параметр	Тип данных	Описание
Доступные значения	<ul style="list-style-type: none"> <li>• Строка</li> <li>• Локализуемая строка</li> <li>• Целое число</li> <li>• Дробное число</li> <li>• Дата и время</li> </ul>	Данный параметр даёт возможность задать список заранее определенных значений пользователю.
Значение по умолчанию	<ul style="list-style-type: none"> <li>• Строка</li> <li>• Локализуемая строка</li> <li>• Логическая переменная</li> <li>• Целое число</li> <li>• Дробное число</li> <li>• Дата и время.</li> </ul>	Данный параметр даёт возможность задать значение по умолчанию. При активном параметр «Множественные значения» будет доступен выбор из введённых значений.
Автозаполняемое значение	<ul style="list-style-type: none"> <li>• Строка;</li> <li>• Текст;</li> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число</li> </ul>	Поле ввода формулы для вычисления значения, которая задается в виде valuePath. (См Приложение 1) и поддерживаемых формул указанных в приложении 2.

	<ul style="list-style-type: none"> <li>• Вычисляемое дробное число</li> <li>• Вычисляемая дата и время</li> <li>• Вычисляемая дата</li> </ul>	
Название типа	<ul style="list-style-type: none"> <li>• Ссылка</li> <li>• Обратная ссылка</li> <li>• Таблица</li> </ul>	<p>Для типа данных Таблица в выпадающем списке отображаются все типы с категорией «Структура».</p> <p>Для типов данных «Ссылка», «Обратная ссылка» в выпадающем списке отображаются все неабстрактные типы УЕ не имеющие категорию «Структура». Выбранные значения используются как фильтр на какие типы может ссылаться поле либо какие типы попадают в поле с типом данных «Обратная ссылка»</p>

В зависимости от значения, которое присвоит пользователь параметру «Тип данных», могут добавляться следующие ограничения. Описание работы с ограничениями даны в приложении 3

### 3.5. Добавление справочников в конфигурацию

Справочники выполняют две основные функции:

- они хранят и отображают нормативно-справочную информацию (например, список типов закрытых радионуклидных источников);
- на основании записей справочников формируются списки предопределённых значений, которые могут использоваться в остальных разделах данного ПО. Это позволяет не только сократить объем повторно вводимых данных, но и уменьшает вероятность внесения ошибок со стороны пользователя ПО.

Для создания справочника необходимо:

1. Создать тип, наименование которого будет наименованием справочника (см. раздел 3.1 настоящего Руководства).

Чтобы справочник отображался на вкладке «Справочники», необходимо параметр «Абстрактный» сделать неактивным.

2. Присвоить созданному типу категорию «Справочник» (см. раздел 2.2 настоящего Руководства).

3. Заполнить недостающие данные в созданных полях и добавить другие поля при необходимости (см. раздел 3.2 настоящего Руководства).

## 4. ИМПОРТ И ЭКСПОРТ КОНФИГУРАЦИИ

### 4.1. Импорт конфигурации

Импорт конфигурации позволяет импортировать дерево конфигурации и заполненные справочники.

Для импорта конфигурации необходимо:

1. Войти в ПО пользователем с ролью «Настройщик».

2. Нажать на кнопку  в правом верхнем углу рядом с логином пользователя.

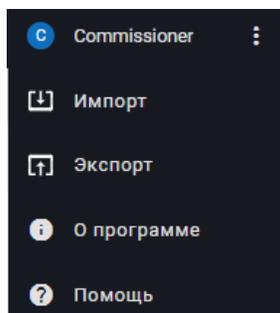


Рисунок 4.1. Подменю после нажатия кнопки 

3. В выпавшем списке нажать кнопку **Импорт**.

4. Если в конфигураторе создавались типы учетных единиц, то появится валидационное сообщение рисунок 4.2.

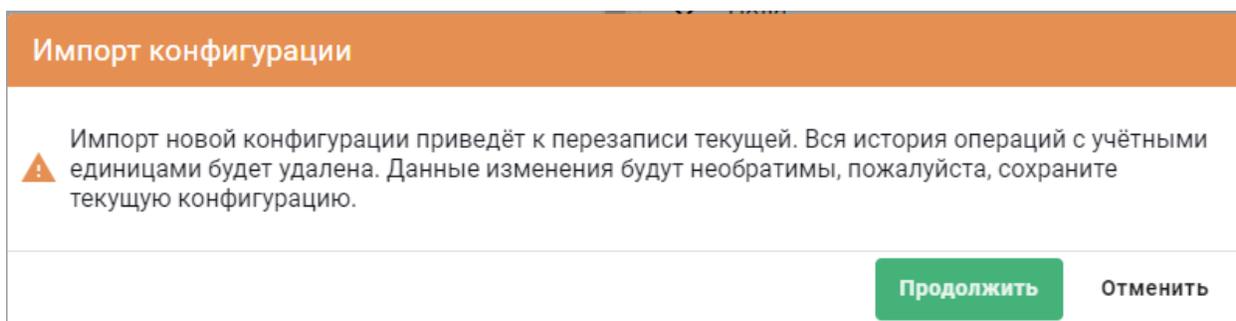


Рисунок 4.2. Валидационное сообщение при импорте конфигурации

5. Нажать **Продолжить**.

6. Выбрать файл в формате **\*.zip**.

После выполнения данных действий появится экспортированная конфигурация.

### 4.2. Экспорт конфигурации

Экспорт конфигурации позволяет экспортировать дерево конфигурации и заполненные справочники.

Для экспорта конфигурации необходимо:

1. Войти в ПО пользователем с ролью «Настройщик».

2. Нажать на кнопку  в правом верхнем углу рядом с логином пользователя.

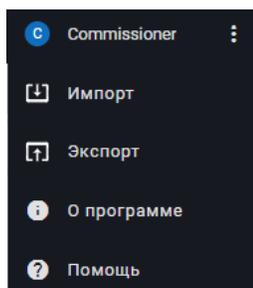


Рисунок 4.3. Подменю после нажатия кнопки 

3. В выпавшем списке нажать кнопку **Экспорт**.

4. Экспортируется файл в формате **\*.zip**.

После выполнения данных действий появится загруженная конфигурация.

## 5. КОНФИГУРИРОВАНИЕ ОПЕРАЦИЙ

Операции являются одной из важнейших особенностей функционала ПО, поскольку посредством выполнения операций осуществляется изменение одной или более характеристик учетной единицы. Настройка набора операций в системе реализуется в Системе гибко. Для настройки операций в ПО должен быть установлен модуль «Учетная единица». Операции и их описание описывается в файле customization.json, который находится в папке с модулем «Учетная единица» (По-умолчанию : C:\Program Files\AtomiSoft\Prima IDC\customization\accounting-unit-module) и содержит следующие данные:

Таблица 5.1. Структура синтаксиса для конфигурирования операции

Параметр	Возможное значение	Описание
"name"	Уникальное название.	Системное название операции
"displayName"	<ul style="list-style-type: none"> <li>Пользовательское значение (перевод журнала на UI не происходит);</li> <li>Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)</li> </ul>	Отображаемое название операции.
"code"	Целое число	Код операции
"group"	<ul style="list-style-type: none"> <li>Пользовательское значение (перевод журнала на UI не происходит);</li> <li>Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)</li> </ul>	<p>Название группы, в которую входит операция.</p> <p>Операции с одинаковым значением в этом параметре объединяются в одну группу на странице добавления новой операции.</p>
"route"	<ul style="list-style-type: none"> <li>"accountingUnit/registration" - операция на основе “Регистрации учётной единицы”</li> <li>"accountingUnit/modification" - операция на основе “Изменение свойств”.</li> <li>"accountingUnit/deregistration" – операция на основе «Снятие с учёта».</li> <li>"accountingUnit/rebatching" - операция на основе “Перегруппировка УЕ по схеме 1-N” или “Перегруппировка УЕ по схеме N-1”</li> </ul>	Указывается базовая (рутовая) операция, на основе которой кастомизируется операция.
"tidSupport"	<ul style="list-style-type: none"> <li>true - действия с УИВ поддерживаются в операции;</li> <li>false - действия с УИВ не поддерживаются в операции</li> </ul>	Поддержка в операции УИВ.

"customization"	Указаны в таблице 5.2	Параметры описывающие операцию
-----------------	-----------------------	--------------------------------

Таблица 5.2. Поддерживаемые параметры в "customization"

Параметр	Возможное значение	Описание
"includedEntityDefinitionIds"	ID типов учётных единиц	Указываются ID типов учётных единиц, которые участвуют в операции.
"services"	Значения из таблицы 5.5.	Параметр, который указывает, что в операции доступна работа сервиса. Доступные сервисы указаны в приложении 9, 10.
"clone"	Значения из таблицы 5.6.	Опциональный параметр, который используется только если "route": "accountingUnit/registration" Параметр, который указывает, что на вкладке «Данные» операции на базе регистрации УЕ, возможно добавить еще одну УЕ на основе данных уже добавленной УЕ в операцию (Создать копию с предзаполненными некоторыми данными).
"multiEdit"	– true - Да; – false - Нет.	Параметр, который используется только для "route": "accountingUnit/modification" или "accountingUnit/deregistration" Параметр, который указывает, используется ли в операции групповое изменить свойства сразу у нескольких УЕ одного типа.
"range"	Значения из таблицы 5.7.	Опциональный параметр, который используется только если "route": "accountingUnit/registration" Параметр, который указывает, что доступна возможность законфигурировать и выполнить операцию на базе регистрации УЕ по заданному диапазону значений полей, в результате которой создается количество записей, соответствующих заданному диапазону.
"entityMode"	"automated"	Опциональный параметр, указывающий на то, что выбор учётной единицы для операции будет происходить с помощью поля по автоматизированному выбору учётной единицы (через сканер штрих-кодов).
"automatedMode"	Значения из таблицы 5.4.	Обязательный параметр при наличии в операции параметра "entityMode": "automated".

		Параметр automatedMode описывает условие для работы <u>автоматизированного контроля</u> с выбором УЕ, которая будет участвовать в операции.
"includedFields"	Названия полей для учетных единиц	Указываются необходимые поля для всех типов УЕ, которые участвуют в операции.  Если в типе УЕ отсутствует поле, указанно в перечислении, то такое поле не отображается в операции с таким типом УЕ.
"phantomFields"	Описание фантомных полей	Описывается участие фантомных полей в операции.
"includedCategories"	name категории	УЕ, на которых присвоена указанная категория будут участвовать в операции.
"excludedCategories"	name категории	УЕ, на которых присвоена указанная категория не будут участвовать в операции.
"excludedFields"	Массив с названием полей типа УЕ	Указанные названия полей будут исключены из операции.

Таблица 5.3. Описание параметра includedFields.

Параметр	Возможное значение	Описание
"name"	Название поля	Указывается системное название поля, которое будет использоваться в операции.
"required"	<ul style="list-style-type: none"> <li>• true - Да;</li> <li>• false - Нет.</li> </ul>	Указывает на обязательность заполнения поля в операции, т.е. значение поля не может быть null (пустым).
"disabled"	<ul style="list-style-type: none"> <li>• true - Да;</li> <li>• false - Нет.</li> </ul>	Опциональный параметр, который указывает будет ли поле активное или неактивное.
"defaultValue"	<ul style="list-style-type: none"> <li>• константа*;</li> <li>• null -пустое значение;</li> </ul>	<p>Значение по умолчанию в поле</p> <p>*Если в поле со ссылкой необходимо по умолчанию выбрать какую-либо учетную единицу, то в параметре "defaultValue" добавляется два элемента:</p> <p>"entityId" - указывается ID типа, УЕ которого необходимо выбрать по умолчанию</p> <p>"entityDefinitionId": //ID учетной единицы, которая принадлежит типу УЕ, указанному в параметре "entityId"</p>

"mode"	"automated"	Опциональный параметр, указывающий на то, что выбор учётной единицы в поле будет происходить с помощью автоматизированного контроля.
"automatedMode"	Значения для из таблицы 5.4.	Обязательный параметр при наличии в операции параметра "entityMode": "automated".  Параметр automatedMode описывает условие для работы автоматизированного контроля.
"visibleFields"	список полей: Например: ["name", "status", ...]	Опциональный параметр только для полей с типом данных «Ссылка». Указывает какие поля будут по умолчанию отображаться в таблице pop-up выбираемых ссылок. Если не указать будут отображаться все поля.

В каждом параметре automatedMode добавляется элемент targetFields, в котором указывается список типов УЕ и соответствующее им настоящее (target) поле, в котором хранится уникальное значение для идентификации УЕ. Каждое настоящее (target) поле описывается следующими характеристиками:

Таблица 5.4. Описание параметра automatedMode, задающего условие работы автоматизированного контроля выбора учётной единицы.

Параметр	Описание
"entityDefinitionId"	ID типа УЕ, в котором будет производиться сравнение считанного значения через оборудование.
"fieldName"	Системное имя target поля, по которому будет произведен поиск УЕ. В указанном поле хранится уникальное значение идентифицирующее УЕ, которое сравнивается со считанным значением с оборудования.

Например:

```
"automatedMode": {
  "targetFields": [
    {
      "entityDefinitionId": 35, //ID типа УЕ, в данном случае справочник
      "Сотрудники"
      "field": "individualNumber" //поле, по которому будет произведен поиск
      сотрудника в автоконтроле (поле «Индивидуальный номер»)
    },
    {
      "entityDefinitionId": 16,
      "field": "passportNumber"
```

```

}
]
}

```

Таблица 5.5. Описание параметра "services"

Параметр	Значение параметра	Описание
"systemName"	Уникальное название сервиса	Уникальное название сервиса, который будет вызван в операции
"type"	<ul style="list-style-type: none"> <li>• "button" - сервис будет вызываться с помощью кнопки;</li> <li>• "file" - сервис будет вызываться с помощью кнопки, которая поддерживает загрузку файла.</li> </ul>	Тип контроля, который будет инициировать вызов работу сервиса в операции.
"displayName"	<ul style="list-style-type: none"> <li>• Пользовательское значение (перевод контроля для вызова сервиса на UI не происходит);</li> <li>• Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)</li> </ul>	Отображаемое название контроля для вызова сервиса в операции.
"tooltip"	<ul style="list-style-type: none"> <li>• Пользовательское значение (перевод всплывающей подсказки на UI не происходит);</li> <li>• Ключ для перевода (перевод всплывающей подсказки отображается согласно значениям из файла локализации)</li> </ul>	Всплывающая подсказка для контроля по вызову сервиса
"entityDefinitionIds"	ID типа UE	ID типов UE, для которых возможно будет инициировать вызов сервиса в операции.
"prerequisites"	[         {           "entityDefinitionId": ID справочника,           "requiredFields": ["название поля"]         },       ]	Параметр отвечает за передачу в сервис необходимых справочников и их полей для успешной работы сервиса.
"script"	"bin/parsers/название Python-скрипта.py"	Данный параметр используется только для "type": "button".

"fileExtensions"	[ {"extension": ".расширение файла", "parser": "bin/parsers/название Python-скрипта.py"}, ]	Данный параметр используется только для "type": "file". Указывается для каждого расширения файла, которые могут быть загружены, путь к необходимому Python-скрипту для обработки файла. Доступны следующие Python-скрипты: • для расширения .spe - "bin/parsers/spe2json.py"; • для расширения .ats - "bin/parsers/ats2json.py".
"parameters"	Перечень параметров для работы сервиса: • для сервиса по заполнению данных в операции из загруженного файла – Приложение 9.	Указывается перечень параметров, который необходим для работы сервиса. Перечень зависит от типа вызываемого сервиса. Для успешной работы сервиса все значения в "parameters" заполняются строго последовательно и в соответствии с указанным синтаксисом.

Таблица 5.6. Описание параметра clone

Параметр	Значение параметра	Описание
"entityDefinitionId"	ID типа UE	Указанный ID типа UE, который поддерживает создание копии UE заполненными значениями в полях из уже добавленной UE, указанного типа в операцию на вкладке «Данные».
"excludedFields"	массив названий полей	Указывается массив названий полей, которые не будут заполнены значениями UE, на основе которой происходит их копирование в новую UE.

Например:

```
"customization": {
  ...
  "clone": [
    {
      "entityDefinitionId": 18, // ID типа UE, который поддерживает копирование
      значений полей из уже добавленной UE, указанного типа в операцию на вкладке «Данные»
      "excludedFields": ["name", "inventoryNumber"] // поля, которые не будут
      заполнены значениями UE, на основе которой происходит копирование // includedFields - не
      поддерживаются
    },
    {
      "entityDefinitionId": 16,
      "excludedFields": ["name", "factoryNumber"] // includedFields - не
      поддерживаются
    }
  ],
  ...
}
```

Таблица 5.7. Описание параметра range

Параметр	Значение параметра	Описание
"entityDefinitionId"	ID типа УЕ	Указывается ID типа УЕ, при постановке УЕ которого поддерживается ввод значений в поле по заданному диапазону.
"field"	См. таблицу 5.8	Указывается поле (или массив полей) и его (их) вид заполнения значений по диапазону.

Таблица 5.8. Описание параметра "field" в параметре "range"

Параметр	Значение параметра	Описание
"name"	название поля УЕ	Указывается название поля, в котором добавятся числа из введенного диапазона по заданному виду в параметре "pattern". Поддерживаются только поля с типом данных "Строка" и <b>не множественным значением</b> . Если указано поле с другим типом данных, то система такое поле игнорирует и ввод данных через диапазон для него не применяется.
"pattern"	<ul style="list-style-type: none"> <li>• {value};</li> <li>• {index};</li> <li>• комбинация из {value} и {index}</li> </ul>	<p>Указывается вид добавления чисел из диапазона к значению в указанном поле "name".</p> <p>Все выражения {value} заменяются на значение из поля указанного в "name"</p> <p>Все выражения {index} заменяются на значения из диапазона.</p> <p>Для выражения {index} поддерживается также описатели стандартного формата C#, которые задаются через символ ":" {index:форматтер}</p> <p><i>Например: Диапазон задан от 1 до 3. В поле name введено значение "ABC". Вид добавления указан: "pattern":{value}{index:d4}. Где d{число} - минимальное количество разрядов для диапазона. Результатом будет являться добавление в операцию регистрации 3-х учётных единиц, в которых в поле name будут значения: ABC0001, ABC0002, ABC0003 соответственно, а остальные поля во всех УЕ будут иметь одинаковые значения.</i></p>

Пример указания параметра range в кастомизации операции:

```
"customization": {
...
  "range": [
    {
      "entityDefinitionId": 42,
      "fields": [
        {
          "name": "number",
```

```
    "pattern": "{value}{index:d4}" // index - номер из диапазона, // value -  
    введенное пользователем значение, // d4 - минимальное число разрядов  
  }  
  ]  
}  
...  
}
```

## 6. КОНФИГУРИРОВАНИЕ ЖУРНАЛОВ

Журналы используются для отражения изменений атрибутов, описывающих учётную единицу. Изменение атрибутов (полей) учётной единицы в ПО выполняется через операции. Формирование журнала происходит по изменению какого-либо одного или более триггерного поля, описывающего учётную единицу. Триггерное поле может быть сложным, если первоначальное триггерное поле является ссылкой. (Например: необходимо построить журнал перемещений ИИИ, который лежит в контейнере, а контейнер лежит в сейфе и сейф стоит в помещении. Для такой ситуации перемещение самого ИИИ, или перемещение контейнера с ИИИ, или перемещение сейфа должно инициировать запись в журнале перемещений ИИИ.)

Запись в журналах может отражать только новое состояние учётной единицы, предыдущее состояние отражается в предшествующей записи (-1 от новой) (Например, журнал учета ЗРИ, журнал учета ОРИ). Также запись в журнале может содержать информацию было-стало, т.е. в одной записи журнала есть информация и о новом состоянии учётной единицы, и о предшествующем состоянии учётной единицы (Например журнал перемещений).

В ПО реализована гибкая настройка журналов. Для настройки журналов в ПО должен быть установлен модуль «Отчеты и журналы». Каждый журнал и его описание описывается в файле customization.json, который находится в папке с модулем «Отчеты и журналы» (По-умолчанию: C:\Program Files\Atomisoft\ Prima IDC\customization\reports-module) и содержит следующие данные:

Таблица 6.1. Структура синтаксиса журнала

Параметр	Описание
name	Системное название журнала
displayName	Отображаемое название журнала. Пользовательское значение (перевод журнала на UI не происходит); Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)
entityDefinitionIds	ID типов учётных единиц, которые попадают в журнал

triggerFields	Триггерные поля учетных единиц, изменения которых создают запись в журнале. Структура описана в таблице П.4.1. приложения 4
fields	Поля, которые входят в журнал. Структура описания полей журнала зависит от значений в параметре source. Список поддерживаемых значений описан в таблице П.4.2., П.4.3. приложения 4.
postprocessing	Постпроцессинг - действия с журналом после его формирования. Как правило это фильтрация, сортировка и другие действия над данными журнала. Структура и поддерживаемые типы постпроцессинга описаны в таблице П.4.5. приложения 4.
permissions	Указывается набор прав для журнала. Поддерживаемый набор и структура их описания указан в таблице 6.2.

Таблица 6.2. Синтаксис прав для журнала

Параметр	Описание
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.view",	Право для просмотра журнала.
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.export"	Право для экспорта данных из журнала. При экспорте данных учитываются настройки фильтрации, сортировки, порядка и отображения колонок.

## 7. КОНФИГУРИРОВАНИЕ ОТЧЕТОВ

Отчеты в ПО выполняют функцию подготовки необходимых данных по каким-либо характеристикам и последующий их экспорт в файл excel.

Для работы и настройки отчетов в ПО должен быть установлен модуль «Отчеты и журналы». Каждый отчет и его описание описывается в файле customization.json, который находится в папке с модулем «Отчеты и журналы» (По-умолчанию: C:\Program Files\AtomiSoft\ Prima IDC\customization\reports-module)

Настройка отчётов в Системе делится на две взаимодополняемые части:

1. Описание кастомизации отчета в customization.json.
2. Создание и описание excel шаблона для отчета на основе данных, описанных в кастомизации.

Первым делом Настройщик в кастомизации описывает логику и необходимые поля для отчёта, а затем в шаблоне excel указывает в нужных ячейках название описанных в кастомизации полей, которые будут формировать данные для отчёта.

Кастомизация отчётов содержит следующие данные:

- название отчета;
- id типов УЕ, которые попадают в отчёт; (если не указан ни один id, в выборку не попадает ни один тип УЕ, т.е. отчёт отображается пустой)
- список полей
- список триггерных полей, если отчёт за период. (если триггерные поля не указаны для отчёта, который формируется за определенный период, то отчёт будет пустой);
- Логика сортировки и фильтрации записей (постпроцессинг).

Шаблон excel содержит следующие данные:

- Данные по умолчанию:
  - названия отчета;
  - титульный лист для отчета, например с данным организации;
  - шапка таблицы отчёта;
  - другие дефолтные данные (утверждаю, согласовано и т.д.).
- Указать области данных в шаблоне в соответствии с приложением 5.
- Указание полей описанных в кастомизации в необходимых ячейках, для формирования таблицы отчёта в виде `{{item["name поля из кастомизации"]}}`;
- Необходимые теги ClosedXML (Приложение 6);

- Указать имя файла шаблона excel, который используется для формирования шаблона. После имени файла через точку указывается языковой префикс для определенной локализации. Т.е. при переключении языка на русский для формирования отчета будет использоваться шаблон {имя файла excel}.{ru}.

Процесс реализации логики работы (в системе)

- Найти все учётные единицы по типу, которые должны попасть в отчёт;
- Отфильтровать полученные УЕ:
  - если отчет на определенную дату, то в отчет приходят УЕ на дату отчёта;
  - если отчет за период дат, то для УЕ, находятся изменения по всем триггерным полям;
- Отфильтровать и отсортировать УЕ по правилам, заданным в кастомизации модуля (например: активность, категории);
- Выполнить логику сложных полей;
- Передать данные в шаблон excel и расположить их в необходимых ячейках согласно названиям полей в шаблоне.
- Сформировать отчет.

Таблица 7.1. Структура синтаксиса отчета:

Параметр	Возможные значения	Описание
name	пользовательское значение	Название отчёта с использованием пользовательского значения. Название отчета на UI не переводится при изменении локализации.
	ключ для перевода	Название отчёта с использованием ключа для перевода. Название отчета на UI переводится при изменении локализации. Перевод ключа отображается согласно значениям из файла локализации.
type	пользовательское значение	Тип отчёта.
template	{имя файла excel без языкового префикса} Например «ORS PIL».	Имя файла шаблона excel (без языкового префикса), который используется для формирования отчета.
usePeriod	<ul style="list-style-type: none"> <li>• false - отчёт формируется на конкретную дату</li> </ul>	Указывается как формируется отчёт: на конкретную дату или за определённый период.

	<ul style="list-style-type: none"> <li>• true - отчёт формируется за определённый период;</li> </ul>	
entityDefinitionIds	ID типов УЕ	Указанные ID типов учётных единиц, которые попадают в отчёт.
triggerFields	Таблица П.4.1. Приложение 4	Триггерные поля учетных единиц, изменения которых создают запись в отчёте. Используется только для отчетов, которые готовятся за период (параметр "usePeriod" = true)
fields		Поля, которые входят в журнал. Структура описания полей отчёта зависит от значений в параметре source. И указана в Таблицах П.4.2, П.4.3. Приложения 4.
postprocessing		Постпроцессинг - действия с отчётом после его формирования. Как правило это фильтрация, сортировка и другие действия над данными отчёта. Структура и поддерживаемые типы постпроцессинга описаны в Приложении 4, таблица П.4.5.
permissions	Таблица 7.2.	Указывается набор прав для отчёта. Поддерживаемый набор и структура их описания указана в таблице 7.2.

Таблица 7.2. Синтаксис прав для отчёта

Параметр	Описание
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.view",	Право для просмотра отчёта.
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.generate"	Право на создание отчёта.
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.download"	Право на скачивание отчёта.
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.remove"	Право на удаление отчёта.
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.send"	Право на отправку отчёта.

## 8. ВЕДЕНИЕ СПРАВОЧНИКОВ

### 8.1. Заполнение справочников

1. В главном меню выбрать раздел «Справочники».
2. Выбрать необходимый справочник для заполнения.
3. Нажать кнопку **+Добавить**.
4. Заполнить необходимые поля.
5. Нажать **Сохранить**.

Для отмены ввода данных необходимо нажать кнопку **Отменить**.

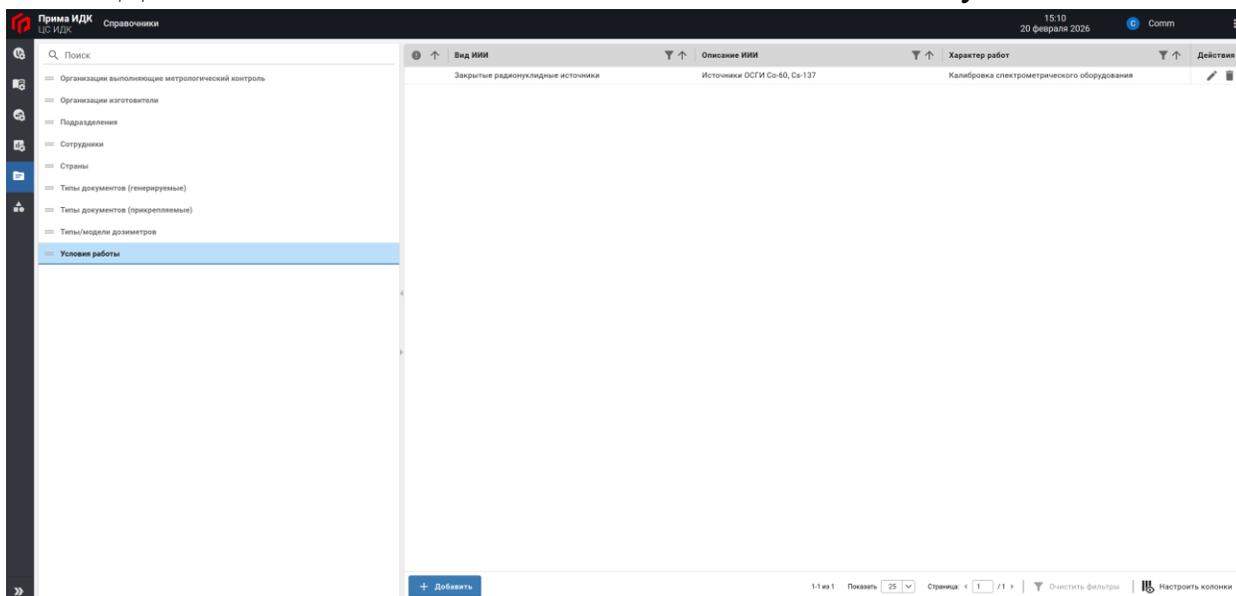


Рисунок 8.1. Пример созданных справочников в конфигурации по учету источников ионизирующего излучения

### 8.2. Редактирование справочников:

1. Перейти на вкладку «Справочники».
2. Выбрать справочник, в котором необходимо отредактировать объект.
3. Для редактируемого объекта в колонке «Действия» нажать кнопку **Редактировать** (  ).
4. Отредактировать необходимые значения.

При редактировании справочников появляется дополнительный параметр 

Дата*	01.01.1970	×	Время*	03:00	×	
-------	------------	---	--------	-------	---	---

. Параметр «Дата и Время» в справочниках отвечает за актуальность редактируемого объекта на определённый момент времени.

### 8.3. Удаление объекта из справочника:

1. Перейти на вкладку «Справочники».

2. Выбрать справочник, в котором необходимо удалить объект.

3. Для удаляемого объекта в колонке «Действия» нажать кнопку

**Удалить** (  ).

4. Для подтверждения действия нажать кнопку **Сохранить** (  ).

Для отмены удаления необходимо нажать кнопку **Отменить** (  ).

## 9. ДЕЙСТВИЯ ПРИ АВАРИЙНЫХ СИТУАЦИЯХ

ПО должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями администратора, неверным форматом или недопустимыми значениями входных данных. В указанных случаях администратору выдаются соответствующие аварийные сообщения, после чего ПО возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных. Аварийные ситуации могут возникать как из-за ошибок в программных продуктах, так и из-за неправильной настройки.

Основными признаками аварийной ситуации являются:

1. Отсутствие на экране необходимой страницы.
2. Окна с сообщениями о нештатной ситуации.

При отказе магнитных носителей или обнаружения ошибок в данных администратор ПО должен восстановить файлы и данные, необходимые для корректной работы ПО из последней резервной копии. Если администратор не может устранить ошибки в данных, следует обратиться к разработчику ПО. При этом необходимо указать перечень данных, содержащих ошибки и правильные значения искаженных атрибутов

В случае возникновения других аварийных ситуаций при работе с ПО и невозможности устранить их с помощью средств администрирования, системы управления базой данных, операционной системы следует обратиться к разработчику ПО. При этом необходимо описать признаки аварийной ситуации и действия, которые были выполнены пользователем непосредственно перед возникновением аварийной ситуации. Ниже описаны основные возможные аварийные ситуации и способы их решения.

<b>Аварийная ситуация</b>	<b>Возможные потери информации</b>	<b>Способ ликвидации последствий</b>	<b>Исполнитель</b>
Отключение питания аппаратных средств	Несохраненные пользователем данные	Повторный ввод и сохранение информации	Пользователь
Выход из строя аппаратных средств (за исключением жесткого диска)	Несохраненные пользователем данные	Повторный ввод и сохранение информации	Пользователь
Сбой при передаче данных	Передаваемая информация	Повторная отправка данных на сервер	Пользователь

<b>Аварийная ситуация</b>	<b>Возможные потери информации</b>	<b>Способ ликвидации последствий</b>	<b>Исполнитель</b>
Отсутствие на экране необходимой страницы	Несохраненные пользователем данные	Перезагрузка страницы кнопкой «Обновить» интернет-браузера; возврат на предыдущую страницу и повторный клик по ссылке на необходимую страницу	Пользователь
Окна с сообщениями о нештатной ситуации	Несохраненные пользователем данные	Выполнить рекомендации, указанные в сообщении, если таковые имеются. При необходимости обратиться к администратору.	Пользователь

## Описание работы с valuePath

valuePath – формат описания запроса, используемый для доступа к полям в пределах учётной единицы и ее ссылок (как прямых, так и обратных) на другие учётные единицы.

Язык запросов построенный по следующей схеме:  
`[$field2.fieldB.fieldC1...]`

\$ - указывает, что будет использовать запрос в виде valuePath.

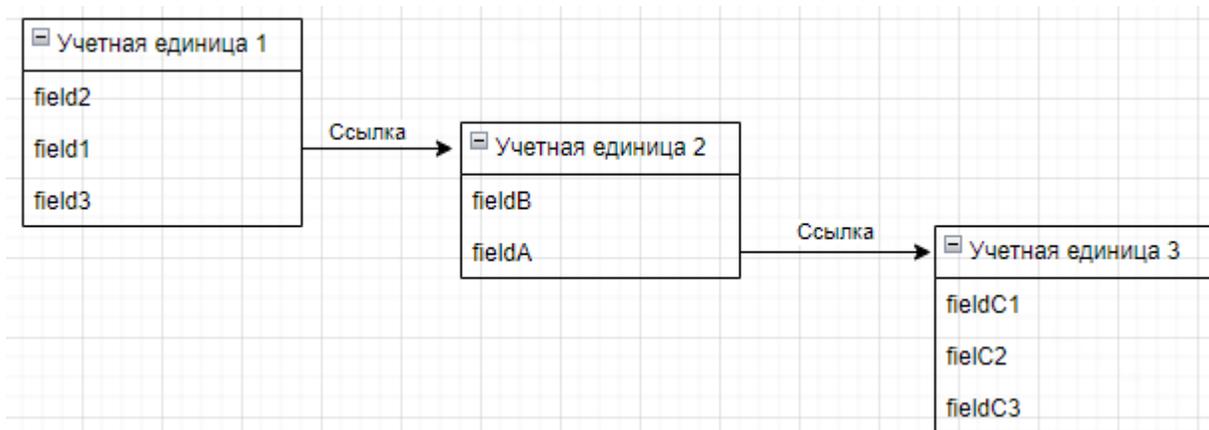
*field1* - имя поля в пределах одной учётной единицы (если в поле field тип данных является ссылкой на какую-либо учётную единицу, то после него ставится точка и указывается необходимое поле fieldA в учётной единице, на которую ссылаются).

*fieldA* - имя поля в учётной единице, на которую ссылается field1.

*fieldC1* - имя поля в учётной единице на который ссылается fieldA.

...

*Пример1:* Учетная единица 1 имеет поле *field2*, *field3* и *field1*, которое ссылается на другую учётную единицу с полями *fieldB*, *fieldA*, в которой *fieldA* ссылается справочник C1. В справочнике C1 есть поле *fieldC1* и *fieldC2*. Необходимо задать ограничение для поля *field1* учётной единицы 1, которое  $\geq$  *fieldC1* из справочника C1, на который ссылается *fieldA*, учётной единицы B.



Результат записи ограничения: **оператор:** “ $\geq$ ” ; **значение:** `[$field1.fieldA.fieldC1]`.

Уровень вложенности не ограничен. На конце valuePath должно быть поле нессылочного типа.

Выражения valuePath можно использовать в качестве аргументов в операторах и функциях (описаны в приложении 2). Так же, как и константы, и результаты выполнения других функций.

*Пример2:*

A - учетная единица. linkToB - поле в учетной единице A с типом данных ссылка, содержащая ссылку на УЕ B

B - учетная единица. reverseLinkToC - поле в учетной единице B с типом данных обратная ссылка, содержащая ссылку на единственную УЕ C

C - учетная единица. tableD - поле в учетной единице C с типом данных таблица, содержащая единственную запись - УЕ D

D - структура. stringField - поле в структуре D с типом данных строка, содержащее значение "HelloWorld!".

Формула: [\$linkToB.reverseLinkToC.tableD.stringField]

Результат: "HelloWorld!"

### Синтаксис поддерживаемых операторов и формул

Для одного поля учётной единицы задается одно значение выражения (формулы). Выражение может состоять из функций и/или операторов, а также констант.

Аргументами в функциях и для используемых операторов могут быть как константы, так и значения полей, которые указываются в виде `valuePath` (правила указания `valuePath` см. в приложении 1). Значения полей, используемых в выражении, могут быть разных типов данных. Все значения в выражении перед вычислениями приводятся к типу данных первого аргумента.

Например:

- `"abc" + 12 = "abc12"` (строка + число = строка + строка);
- `"1" + "2" = 12` (строка + строка = строка + строка);
- `"" + 4 + 5 = "45"` (пустая строка + число + число = строка + строка + строка);
- `1 + 2 = 3` (число + число = число + число);
- `5 + "6" = 11.` (число + строка = число + число).

При отсутствии значения в одном из полей при расчете выполняется следующая логика:

- строка, текст - `Option.None == ""` (пустая строка);
- целое число и дробное число - `Option.None == 0`;
- дата и время - `Option.None == ошибка`.

После выполнения расчета выражения результат приводится к типу данных поля, в котором задано значение для автозаполнения (выражение `valuePath`). Если результат привести невозможно, то система возвращает ошибку проведения расчёта и расчётное значение будет отсутствовать.

Поле, может являться элементом `valuePath` для другого вычисляемого поля. Пример: `fieldTest1 = 5, fieldTest2 = 6, fieldTest3 = 20. fieldTest4 = Sum([$fieldTest1], [$fieldTest2], [$fieldTest3]); fieldTest5= fieldTest4 +29`

- Результат: `fieldTest4 = 31; fieldTest5= 50.`

### П.2.1. Поддерживаемые операторы в выражениях

Оператор	Описание	Тип данных результата	Пример
+ (Сложение)	При сложении двух строк происходит континуация (продолжение) строк. Задается в виде: {field1} + {field2}. Если необходимо указать разделитель между складываемыми полями выражение задается в виде: {field1} + 'знак разделителя' + {field2}.	Строка	fieldTest1 = Строка1; fieldTest2= Строка2 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: Строка1Строка2.
	При сложении двух текстовых полей происходит континуация (продолжение). Задается в виде: {text1} + {text2}. Если необходимо указать разделитель между складываемыми полями выражение задается в виде: {text1} + 'знак разделителя' + {text2},	Текст	textTest1 = Текст1; textTest2 = Текст2 Пример: [\$textTest1] + [\$textTest2]. Результат: Текст1Текст2.
	Выполняется сложение целых чисел.	Целое число	fieldTest1 = 4; fieldTest2= 5 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: 9.
	Выполняется сложение дробных чисел.	Дробное число	fieldTest1 = 4,2; fieldTest2= 5,3 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: 9,5.
- (Разность)	Выполняется нахождение разности целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [\$fieldTest1] - [\$fieldTest2]. Результат: 4.
	Выполняется нахождение разности дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [\$fieldTest1] - [\$fieldTest2]. Результат: 6,1.

* (Умножение)	Выполняется умножение целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [fieldTest1] * [fieldTest2]. Результат: 12.
	Выполняется умножение дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [fieldTest1] * [fieldTest2]. Результат: 9,62.
/ (Деление)	Выполняется деление целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [fieldTest1] * [fieldTest2]. Результат: 12.
	Выполняется умножение дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [fieldTest1]/ [fieldTest2]. Результат: 9,62.

## П.2.2. Поддерживаемые функции в выражениях

### П.2.2.1. Общие функции доступа к данным

#### П.2.2.1.1. Функция: selectValues(valuePath, (optional) function)

Функция выравнивает все вложенные массивы в один уровень, находит значение по указанному valuePath для каждого элемента, и выполняет функцию function для каждого элемента получившегося массива

Результат функции используется как параметр в других функциях, которые принимают массив в качестве параметра: concat, count, sum, join, contains, first, max, min, average, selectValues.

Возвращаемое значение: тип данных возвращаемого значения - массив элементов

## Параметры:

1. `valuePath` - содержит `valuePath`, по которому осуществляется поиск значений для результирующего массива
2. `function` - функция, которая будет вызвана для каждого элемента получившегося массива
  - Параметр опционален; если пропущен - дополнительная функция не вызывается
  - Внутренняя функция содержит параметры; параметр такой функции, вместо которого будет передаваться элемент массива, помечается нижним подчеркиванием (см. пример 3 ниже)

### Примеры:

#### Пример 1

`fieldTest2` - ссылка на 2 УЕ, у которых поле `name` содержит значения “Привет” и “Мир”.

Формула: `selectValues([$fieldTest2.name])`

Результат: массив, состоящий из двух элементов [“Привет”, “Мир”]

#### Пример 2

`fieldTest3` - обратная ссылка, которая содержит 2 УЕ у которых поле `field1` содержит значения “Ура” и “Круто”.

Формула: `selectValues([$fieldTest3.field1])`

Результат: массив, состоящий из двух элементов [“Ура”, “Круто”]

#### Пример 3

`radionuclideComposition` - таблица, которая состоит из 2 записей:

element	activity	date
Elem1	13.123456789	2024-12-10 12:00
Elem2	42.424242424	2024-12-14 22:00

Формула: `join(selectValues([$radionuclideComposition.activity], 'round(_, 2)'), '; ')`

Результат: “13.12; 42.42”

#### Пример 4

Формула: `join(selectValues([$radionuclideComposition.date], 'dateFormat(_, \'dd.MM.yyyy HH:mm\')'), '; ')`

Результат: “10.12.2024 12:00; 14.12.2024 22:00”

#### П.2.2.1.2. Функция: `if(condition, returnIfTrue, returnIfFalse)`

Функция позволяет выполнять логические сравнения значений и ожидаемых результатов. Она проверяет условие и в зависимости от его истинности возвращает результат. Поэтому у функции возможны два результата. Первый результат возвращается в случае, если сравнение истинно, второй - если сравнение ложно.

Тип данных возвращаемого значения совпадает с типом данных поля (либо результата вычисления)

При записи условия в функции можно использовать операторы `=`, `!=`, `<`, `<=`, `>`, `>=`, `or`, `and`.

В функции поддерживается составление сложных выражений, т.е значение параметров истина/ложь может также задаваться с помощью функции `if`.

Элементами данной функции, могут являться другие функции.

Параметры:

- `condition` - условие
- `returnIfTrue` - возвращаемое значение (либо формула), которая вернется если условие выполняется
- `returnIfFalse` - возвращаемое значение (либо формула), которая вернется если условие не выполняется

Примеры

*Пример 1*

`fieldTest1 = 30`

`fieldTest2 = "Норма"`

`fieldTest3 = "Отклонение"`

`fieldTest4.code = 20`

Формула: `if ([fieldTest1] > [fieldTest4.code], if ([fieldTest1] > 35, [fieldTest2], [fieldTest3]), [fieldTest3])`

Результат: "Отклонение"

#### П.2.2.1.3. Функция: `getEntityDefinitionId(valuePath)`

Функция возвращает `EntityDefinitionId` для искомого `valuePath`

Возвращаемое значение:

Тип данных возвращаемого значения - целое число

В случае если значение не найдено - вернется -1

Параметры:

- `valuePath` - `valuePath` к полю

При указании пути необходимо указать какое-либо нессылочное поле в последнем сегменте `valuePath`.

Примеры

*Пример 1*

Формула: `getEntityDefinitionId([$name])`

Результат: `EntityDefinitionId` текущего Entity, если у него присутствует поле `name`

*Пример 2*

Формула: `getEntityDefinitionId([$location.name])`

Результат: `EntityDefinitionId` того объекта который находится по ссылке `location` у текущего Entity, если у того присутствует поле `name`

*Пример 3*

Формула: `getEntityDefinitionId([$nonExistingField])`

Результат: -1, так как поля с именем `$nonExistingField` не существует

#### П.2.2.1.4. Функция: `getOperationInfo('dateTime' | 'code')`

Функция для возвращает информацию о последней операции на выбранный момент времени, в рамках которой было изменено значение поля.

Возвращаемое значение:

- Тип данных возвращаемого значения - дата и время в случае, когда запрашивались дата и время проведения операции (`dateTime`)
- Тип данных возвращаемого значения - строка в случае, когда запрашивался код операции (`code`)

Параметры:

Функция принимает один из двух возможных параметров:

- `dateTime` - возвращает дату и время, когда было произведено данное действие
- `code` - возвращает код операции в которой было произведено данное действие

#### П.2.2.2. Функции для работы с массивами

##### П.2.2.2.1. Функция: `concat(param1, param2, ...)`

Функция выполняет конкатенацию всех параметров

Параметры:

- `paramX` - константа, либо `valuePath` к полю, либо результат выполнения функции
  - В функцию может быть передано несколько аргументов, каждый из которых может быть константой, `valuePath` на множественное или единственное значение или массив (например результат выполнения функции `selectValues()`); параметры множественных типов будут переданы в функцию как несколько параметров.
  - Перед конкатенацией, все параметры будут приведены к типу строка. Параметры множественных типов будут переданы в функцию как несколько параметров.
  - Для параметров с типами `ссылка`, `обратная ссылка`, `таблица` используйте `valuePath` либо функцию `selectValues()` для доступа к внутренним полям учетной единицы. Правила приведения полей с типами `ссылка`, `обратная ссылка`, `таблица` к типу строка не определены.

Возвращаемое значение:

Тип возвращаемого значения - строка

Примеры:

*Пример 1*

`fieldTest1` = поле с типом данных `целое число` содержащее значение 100

`fieldTest2` = поле с типом данных `дробное число` содержащее значение 4,5

`fieldTest3` = поле с типом данных `строка` содержащее значение “Мир”

fieldTest4 = поле с типом данных `текст` содержащее значение “Привет!”

Формула: `concat([$fieldTest4], [$fieldTest3], [$fieldTest1], [$fieldTest2])`

Результат: Привет!Мир1004.5

*Пример 2*

fieldTest1 - таблица, которая состоит из 3 записей и полей:

element	mass
Уран	10.5
Торий	12
Цезий	42

fieldTest2 - ссылка на 2 УЕ, у которых поле `name` содержит значения “Привет” и “Мир”.

fieldTest3 - обратная ссылка, которая содержит 2 УЕ у которых поле `field1` содержит значения “Ура” и “Круто”.

Пример: `concat(selectValues([$fieldTest1.element]), selectValues([$fieldTest2.name]), selectValues([$fieldTest3.field1]))`

Результат: УранТорийЦезийПриветМирУраКруто

#### П.2.2.2.2. Функция: `join(valuePath, separator)`

Функция объединяет элементы массива, найденные по `$valuePath` в строку, с использованием заданного разделителя.

Параметры:

Функция принимает два параметра:

1. `valuePath` - путь к множественному полю

- Перед объединением, все параметры в массиве, найденном по `$valuePath`, будут приведены к типу строка
- Для доступа к внутренним полям учетной единицы используйте функцию `selectValues()`

2. `separator` - строковая константа, которая будет использована в качестве разделителя

Возвращаемое значение:

Тип возвращаемого значения - строка

### П.2.2.2.3. Функция: count(valuePath)

Нахождение количества элементов в заданном массиве

Параметры:

- valuePath - valuePath к множественному полю, либо другой массив (например результат выполнения функции selectValues())

При указании пути необходимо указать какое-либо нессылочное поле в последнем сегменте valuePath

Возвращаемое значение:

Тип данных возвращаемого значения - целое число

Примеры:

*Пример 1*

radionuclides - таблица, которая состоит из 3 записей.

element	name	mass
Уран		20
Торий	Th-207	10
Цезий	Cs-137	45

Формула: count([\$radionuclides])

Результат: 3.

*Пример 2*

Формула: count([\$radionuclides.element])

Результат: 3.

*Пример 3*

Формула: count([\$radionuclides.name])

Результат: 2.

*Пример 4*

fieldTest1 - ссылочное поле (может быть как множественная ссылка, так и обратная ссылка). В поле хранятся ссылки на 5 УЕ.

Формула: `count([$fieldTest1.name])`

Результат: 5.

*Пример 5*

fieldTest1 - строка с 5 множественными значениями.

Формула: `count([$fieldTest1])`

Результат: 5.

#### П.2.2.2.4. Функция: `sum(valuePath1, valuePath2, ...)`

Нахождение суммы значений в заданных массивах

Параметры:

- `valuePathX` - `valuePath` к множественному полю, либо другой массив (например результат выполнения функции `selectValues()`)

- В функцию может быть передано несколько аргументов, каждый из которых может быть массивом или единственным значением; параметры множественных типов будут переданы в функцию как несколько параметров

Возвращаемое значение:

Тип данных возвращаемого значения - дробное число

Примеры

*Пример 1*

radionuclides - таблица, которая состоит из 3 записей и полей:

element	mass
Уран	10.5
Торий	12
Цезий	42

Формула: `sum(selectValues([$radionuclides.mass]))`

Результат: 64,5.

*Пример 2*

fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения:  
100; 150; 200

fieldTest2 = поле с типом данных “Дробное число” содержащее множественное значение в котором 3 значения:  
4.5; 5.12; 54.43

fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”;  
“5.37”; “165.28”

Формула: `sum([$fieldTest1], [$fieldTest2], [$fieldTest3])`

Результат: 686,7.

*Пример 3*

fieldTest1 = поле с типом данных “Целое число” содержащее значение: 100

fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4.5

fieldTest3 = поле с типом данных “Строка” содержащее значение “165.28”

Формула: `sum([$fieldTest1], [$fieldTest2], [$fieldTest3])`

Результат: 269,78.

*Пример 4*

fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения:  
100; 150; 200

fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4.5

fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”;  
“5.37”; “165.28”

fieldTest4 Таблица, которая состоит из 3 записей и полей

element	mass
Уран	10.5
Торий	12
Цезий	42

**Формула:** `sum([$fieldTest1], [$fieldTest2], [$fieldTest3], selectValues('$fieldTest4.mass'))`

**Результат:** 689,65.

#### П.2.2.2.5. Функция: `first(valuePath)`

Функция возвращает первый элемент из массива значений. Выводит всегда один элемент если будет найдено несколько полей или у одного поля будет несколько значений (поле является массивом)

**Параметры:**

- `valuePath` - `valuePath` к множественному полю, либо другой массив (например результат выполнения функции `selectValues()`)

**Возвращаемое значение:**

Тип данных возвращаемого значения совпадает с типом данных найденного поля

**Примеры:**

*Пример 1*

`fieldTest1 = [1, 2, 3]`

**Формула:** `first([$fieldTest1])`

**Результат:** 1.

#### П.2.2.2.6. Функция: `contains(valuePath, expectedValue)`

Функция позволяет проверить, содержит ли массив искомое значение. Целесообразно использовать в условиях.

**Параметры:**

- `valuePath` - `valuePath` к множественному полю, либо другой массив (например результат выполнения функции `selectValues()`)
- `expectedValue` - строка, наличие которой необходимо проверить в массиве

Возвращаемое значение:

Тип данных возвращаемого значения - логическая переменная

Примеры:

*Пример 1*

radiationType - массив строк, содержащий значения 'γ', 'α' и 'β'

Формула: `if(contains([$type.radiationType], 'γ'), 1, 2)`

Результат: 1

*Пример 2*

Формула: `if(contains([$type.radiationType], 'z'), 1, 2)`

Результат: 2

*Пример 3*

Формула: `contains([$type.radiationType], 'z')`

Результат: False

#### П.2.2.2.7. Функция: `arrayContains(stringArray, valuePath)`

Функция позволяет проверить, содержит ли массив искомое значение. Целесообразно использовать в условиях.

Параметры:

- `stringArray` - массив значений в формате '[1, 2, 3]' (набор значений разделенных запятой с пробелом)
- `valuePath` - параметр с `valuePath` или другая функция

Возвращаемое значение:

Тип данных возвращаемого значения - логическая переменная

Примеры:

*Пример 1*

`fieldTest1 = 2.`

Формула: `if(arrayContains('[1, 2, 3]', [$fieldTest1]), 1, 2)`

Результат: 1

*Пример 2*

Формула: `if(arrayContains('[1, 2, 3]', getEntityDefinitionId([$name])), 1, 2)`

*Пример 3*

Формула: `arrayContains('[1, 2, 3]', [$fieldTest1])`

Результат: True

### П.2.2.3. Функции для работы с числами

#### П.2.2.3.1. Функция: `pow(x, y)`

Возведение  $x$  в степень  $y$

Параметры:

- $x$  - число, либо `$valuePath` к числу
- $y$  - число, либо `$valuePath` к числу

Возвращаемое значение:

Тип данных возвращаемого значения - дробное число

Примеры

*Пример 1*

`fieldTest1 = 6;`

Формула: `pow([$fieldTest1], 2)`

Результат: 36

*Пример 2*

`fieldTest1 = 1,4;`

Формула: `pow([$fieldTest1], 0.25)`

Результат: 1.0877573059372771726111171278296

#### П.2.2.3.2. Функция: $\ln(x)$

Нахождение логарифма по основанию  $e$

Параметры:

- $x$  - число, либо `valuePath` к числу

Возвращаемое значение:

Тип данных возвращаемого значения - `дробное число`

Примеры:

*Пример 1*

`fieldTest1 = 2`

Формула: `ln([$fieldTest1])`

Результат: 0.69314718055994530941723212145818

#### П.2.2.3.3. Функция: $\exp(x)$

Возведение  $e$  в степень  $x$

Параметры:

- $x$  - число, либо `valuePath` к числу

Возвращаемое значение:

Тип данных возвращаемого значения - `дробное число`

Примеры

*Пример 1*

`fieldTest1 = 0.5`

Формула: `exp([$fieldTest1])`

Результат: 1.6487212707001281468486507878142

#### П.2.2.3.4. Функция: $\max(\text{param1}, \text{param2}, \dots)$

Функция позволяет найти максимальное значение из массива целых и дробных чисел

Параметры:

- Каждый параметр - число, либо valuePath к числу или массиву чисел

Возвращаемое значение:

Тип данных возвращаемого значения - дробное число

Примеры

*Пример 1*

fieldTest1 - поле с типом данных `целое число` содержащее множественное значение в котором 3 значения: 100; 150; 200

fieldTest2 - поле с типом данных `дробное число` содержащее значение: 4.5

fieldTest3 - поле с типом данных `строка` содержащее множественное значение в котором 3 значения: “2”; “5.37”; “165.28”

fieldTest4 - `таблица`, которая состоит из 3 записей и полей

element	mass
Уран	10.5
Торий	12
Цезий	42

fieldTest5 - `ссылка`, на 2 УЕ, у которых поле `number` содержит значения 4.2 и 345

fieldTest6 - `обратная ссылка`, которая содержит 2 УЕ у которых поле `number` содержит значения 32.87 и 23

Формула: `max([fieldTest1], [fieldTest2], [fieldTest3], selectValues('$fieldTest4.mass'), selectValues('$fieldTest5.number'), selectValues('$fieldTest6.number'))`

Результат: 345

#### П.2.2.3.5. Функция: `min(param1, param2, ...)`

Функция позволяет найти минимальное значение из массива целых и дробных чисел

Параметры:

- Каждый параметр - число, либо valuePath к числу или массиву чисел

Возвращаемое значение

Тип данных возвращаемого значения - дробное число

Примеры

*Пример 1*

fieldTest1 - поле с типом данных `целое число` содержащее множественное значение в котором 3 значения: 100; 150; 200

fieldTest2 - поле с типом данных `дробное число` содержащее значение: 4.5

fieldTest3 - поле с типом данных `строка` содержащее множественное значение в котором 3 значения: “2”; “5.37”; “165.28”

fieldTest4 - таблица, которая состоит из 3 записей и полей.

element	mass
Уран	10.5
Торий	12
Цезий	42

fieldTest5 - ссылка, на 2 УЕ у которых поле `number` содержит значения 4.2 и 345

fieldTest6 - обратная ссылка, которая содержит 2 УЕ у которых поле `number` содержит значения 32.87 и 23

Формула: `min([$fieldTest1], [$fieldTest2], [$fieldTest3], selectValues('$fieldTest4.mass'), selectValues('$fieldTest5.number'), selectValues('$fieldTest6.number'))`

Результат: 2

#### П.2.2.3.6. Функция: `average(param1, param2, ...)`

Функция позволяет найти среднее значение из массива целых и дробных чисел

Расчет среднего значения ведется по всем элементам входящим в формулу. Если поле (включая таблицу или массив), участвующее в расчете пустое, то оно также участвует в общем расчете.

Параметры:

- Каждый параметр - число, либо `valuePath` к числу или массиву чисел

Возвращаемое значение:

Тип данных возвращаемого значения - дробное число

## Примеры

### Пример 1

fieldTest1 - поле с типом данных `целое число` содержащее множественное значение в котором 3 значения: 100; 150; 200

fieldTest2 - поле с типом данных `дробное число` содержащее значение: 4.5

fieldTest3 - поле с типом данных `строка` содержащее множественное значение в котором 3 значения: “2”; “5.37”; “165.28”

fieldTest4 - `таблица`, которая состоит из 3 записей и полей

element	mass
Уран	10.5
Торий	12
Цезий	42

fieldTest5 - `ссылка`, на 2 УЕ у которых поле `number` содержит значения 4.2 и 345

fieldTest6 - `обратная ссылка`, которая содержит 2 УЕ у которых поле `number` содержит значения 32.87 и 23

**Формула:** `average([fieldTest1], [fieldTest2], [fieldTest3], selectValues('$fieldTest4.mass'), selectValues('$fieldTest5.number'), selectValues('$fieldTest6.number'))`

**Результат:** 78.337142857142857

### П.2.2.3.7. Другие функции из библиотеки NCalc

Для работы с числами можно использовать любые функции, описанные в библиотеке NCalc:

<https://ncalc.github.io/ncalc/articles/functions.html>

Параметры:

- Параметры могут содержать как константное значение, так и путь `valuePath`. При передаче параметров в функцию NCalc, значение по пути `valuePath` будет вычислено заранее и передано в функцию библиотеки.
- Можно использовать параметры любых типов данных. При передаче параметров в функцию NCalc, они будут приведены к типу `строка`.

Примеры:

### *Пример 1*

numberField = 42.42

stringField = "13"

Формула: `ceiling([$numberField]) + sign([$stringField]) + 0.5`

Результат: 44.5

## П.2.2.4. Функции для работы с датами

### П.2.2.4.1. Функция: `addDays(dateTime, addingDays)`

Функция для изменения даты на указанное количество дней. Форматирование выполняется в часовом поясе сервера.

Параметры:

- `dateTime` - значение даты и времени, может быть задано в одном из следующих форматов:

- Строковая константа в формате 'dd.ММ.уууу HH:mm' либо другом формате;
- Переменная `[@now]` либо `[@today]`;

○ `valuePath` к полю с типом дата либо дата и время; при наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку.

- `addingDays` - число либо `valuePath` к числу (поле с типом данных целое число либо дробное число)

Возвращаемое значение:

Тип данных возвращаемого значения - дата либо дата и время, в зависимости от типа параметра `dateTime`

Примеры:

### *Пример 1*

fieldTest1 = 01.02.2024 18:30

Формула: `addDays([$fieldTest1], 15)`

Результат: 16.02.2024 18:30

### *Пример 2*

fieldTest1 = 01.02.2024 18:30

Формула: `addDays([$fieldTest1], -1)`

Результат: 31.01.2024 18:30

*Пример 3*

Формула: `addDays([@today], 15)`

Результат: 17.10.2024 00:00

*Пример 4*

Формула: `addDays('16.10.2014 18:30', 1)`

Результат: 17.10.2024 18:30

*Пример 5*

Формула: `addDays('16.10.2014', 1)`

Результат: 17.10.2024 00:00

#### П.2.2.4.2. Функция: `addHours(dateTime, addingHours)`

Функция для изменения даты и времени на указанное количество часов

Параметры:

- `dateTime` - значение даты и времени, может быть задано в одном из следующих форматов:
  - Строковая константа в формате 'dd.ММ.уууу HH:mm' либо другом формате
  - Переменная `[@now]` либо `[@today]`
  - `valuePath` к полю с типом `дата` либо `дата и время`; при наличии в выражении полей других типов, система

расчёт не проводит и возвращает ошибку

- `addingHours` - число либо `valuePath` к числу (поле с типом данных `целое число` либо `дробное число`)

Возвращаемое значение:

Тип данных возвращаемого значения - `дата и время`, форматирование выполняется в часовом поясе сервера

Примеры

### *Пример 1*

fieldTest1 = 01.02.2024 18:30

Формула: `addHours([$fieldTest1], 4)`

Результат: 01.02.2024 22:30

### *Пример 2*

fieldTest1 = 01.02.2024 18:30

Формула: `addHours([$fieldTest1], -1)`

Результат: 01.02.2024 17:30

### *Пример 3*

Формула: `addHours([@today], 15)`

Результат: 17.10.2024 15:00

### *Пример 4*

Формула: `addHours('16.10.2014 18:30', 1)`

Результат: 16.10.2024 19:30

## П.2.2.4.3. Функция: `dateFormat(dateTime, format)`

Функция для форматирования даты и времени в строку

Параметры:

- `dateTime` - значение даты и времени, может быть задано в одном из следующих форматов:

- Строковая константа в формате 'dd.ММ.уууу НН:мм' либо другом формате
- Переменная `[@now]` либо `[@today]`

○ `valuePath` к полю с типом `дата` либо `дата и время`; при наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку

- `format` - формат преобразования в строку; спецификация формата - <https://learn.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings>

Возвращаемое значение:

Тип данных возвращаемого значения - строка, форматирование выполняется в часовом поясе сервера

Примеры

*Пример 1*

fieldTest1 = 01.02.2024 18:30

Формула: `dateFormat([$fieldTest1], 'dd/ММ hh:mm')`

Результат: "01/02 18:30"

#### П.2.2.4.4. Функция: [`@now`]

Переменная, которая автоматически заполняет поле текущей датой и текущим временем

Параметры: отсутствуют

Возвращаемое значение:

Тип данных возвращаемого значения - дата и время

Примеры

*Пример 1. Текущее число 30.05.2024 время 12:43*

Формула: [`@now`]

Результат: 30.05.2024 12:43

#### П.2.2.4.5. Функция: [`@today`]

Переменная, которая автоматически заполняет поле текущей датой и временем 00:00

Для поля с типом данных `дата` устанавливается только значение текущей даты (время игнорируется)

Параметры: отсутствуют

Возвращаемое значение:

Тип данных возвращаемого значения - дата

Примеры

*Пример 1*

Формула: `[@today]`

Результат: 30.05.2024

#### П.2.2.4.6. Функция: `subtractDate(dateTime1, dateTime2)`

Вычитание `dateTime2` из `dateTime1`

Параметры:

- `dateTime1` - значение даты и времени, может быть задано в одном из следующих форматов:

- Строковая константа в формате 'dd.ММ.уууу НН:мм' либо другом формате,

- Переменная `[@now]` либо `[@today]`

- `valuePath` к полю с типом `дата` либо `дата и время`; при наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку

- `dateTime2` - значение даты и времени, может быть задано в одном из следующих форматов:

- Строковая константа в формате 'dd.ММ.уууу НН:мм' либо другом формате

- Переменная `[@now]` либо `[@today]`

- `valuePath` к полю с типом `дата` либо `дата и время`; при наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку

Возвращаемое значение:

Тип данных возвращаемого значения - `целое число`, возвращаемое значение - в секундах

Примеры

*Пример 1*

x - поле типа `дата и время` со значением 2024-10-10 12:00:00

y - поле типа `дата и время` со значением 2024-10-05 12:00:00

Формула: `subtractDate([$x], [$y])`

Результат: 432000

*Пример 2*

x - поле типа дата и время со значением 2024-10-10 12:00:00

y - поле типа дата и время со значением 2024-10-05 10:00:00

Формула: `subtractDate([$x], [$y])`

Результат: 439200

#### П.2.2.5. Функции для рекурсивного доступа к полям и получение предыдущего значения

##### П.2.2.5.1. Функция: `previousValue(valuePath)`

Возвращает предыдущее значение поля, относительно времени, на которое запрашиваются данные

Будет использовано значение по выбранному пути, которое было на 10мс назад от даты записи текущего значения. Функция корректно обрабатывает только в журналах и в генерируемых документах. В остальных местах функция вернет текущее значение поля.

Параметры:

- `valuePath` - `valuePath` к искомому полю

Возвращаемое значение:

Тип данных возвращаемого значения совпадает с типом данных искомого поля

Примеры:

```
previousValue([$specificActivity1])
```

```
previousValue([$container.specificActivity1])
```

##### П.2.2.5.2. Функция: `recursiveJoin(linkFieldName, targetFieldName, separator, deepLevel)`

Объединение в одну строку, через заданный разделитель, значений найденных с помощью рекурсии по ссылкам

Если необходимо указать несколько названий поля - то названия перечисляются через "|"

Параметры:

- `linkFieldName` - название ссылочного поля
- `targetFieldName` - название поля из УЕ, найденной по ссылке, значение которого нужно вернуть
- `separator` - разделитель

- `deepLevel` - глубина рекурсивного поиска

Возвращаемое значение:

Тип данных возвращаемого значения - строка

Примеры:

Нужно отобразить имя местоположения, при этом имена контейнеров - это поле `name`, а имена ячеек - `cellName`

```
recursiveJoin('location', 'name|cellName', '; ', '5')
```

### П.2.2.5.3. Функция: `previousRecursiveJoin(linkFieldName, targetFieldName, separator, deepLevel)`

Запись в одну строку, через заданный разделитель, предыдущих значений полей, найденных с помощью рекурсии по ссылкам

Если необходимо указать несколько названий поля - то названия перечисляются через '|'

Будет использовано значение по выбранному пути, которое было на 10мс назад от даты записи текущего значения. Функция корректно отрабатывает только в журналах и в генерируемых документах. В остальных местах функция вернет текущее значение поля.

Параметры:

- `linkFieldName` - название ссылочного поля
- `targetFieldName` - название поля из УЕ, найденной по ссылке, значение которого нужно вернуть
- `separator` - разделитель
- `deepLevel` - глубина рекурсивного поиска

Возвращаемое значение:

Тип данных возвращаемого значения - строка

Примеры:

Нужно отобразить имя предыдущего местоположения, при этом имена контейнеров - это поле `name`, а имена ячеек - `cellName`

```
previousRecursiveJoin('location', 'name|cellName', '; ', '5')
```

#### П.2.2.5.4. Функция: recursiveDefinitionFind(linkFieldName, condition, entityDefinitionId, returnFieldNameIfTrue, defaultValue, deepLevel)

В ссылочном поле `linkFieldName` осуществляется рекурсивный поиск до тех пор, пока не выполнится условие (`entityDefinitionId` равен либо не равен заданному значению). Если такая УЕ найдена, возвращается значение поля `returnFieldNameIfTrue` найденной УЕ. Если не найдено УЕ, удовлетворяющей условию - возвращается `defaultValue`. Глубина поиска ограничена `deepLevel`.

##### Параметры

- `linkFieldName` - название ссылочного поля, по которому идет поиск
- `condition` - условие = или !=
- `entityDefinitionId` - ожидаемый ID типа учётной единицы
- `returnFieldNameIfTrue` - поле найденной УЕ, значение которого возвращается при истинности условия
- `defaultValue` - значение по умолчанию
- `deepLevel` - глубина рекурсивного поиска

Возвращаемое значение:

Тип данных возвращаемого значения совпадает с типом данных поля `returnFieldNameIfTrue`

Примеры:

```
recursiveDefinitionFind('location', '=', 116, 'fullName', '-', 10)
```

#### П.2.2.6. Форматы дат

В функциях, которые принимают в качестве параметра дату либо дату и время, допускается вводить данные в виде строковых констант. Поддерживаются следующие форматы:

- 'dd.ММ.уууу НН:mm', например '25.10.2024 19:00'
- 'dd.ММ.уууу', например '25.10.2024'
- 'уууу-ММ-dd НН:mm', например '2024-10-25 19:00'
- 'уууу-ММ-dd', например '2024-10-25'

- 'MM/dd/yyyy HH:mm', например '10/25/2024 19:00'
- 'MM/dd/yyyy', например '10/25/2024'
- 'yyyy-MM-ddTHH:mm:ss.fffZ', например '2024-10-25T18:00:00.000Z'

Правила установки ограничений для поля

Для одного поля учётной единицы можно задать одно и более ограничений. При добавлении двух и более ограничений логической связкой таких ограничений, могут являться значение “И” либо “ИЛИ” (т.е. если задать два ограничения для поля со связкой “И”, то значение которое вводится в поле должно удовлетворять двум заданным ограничениям; если задать два ограничения для поля со связкой “ИЛИ”, то значение которое вводится в поле должно удовлетворять одному из двух заданным ограничениям).

Ограничение для поля учётной единицы задаются с помощью двух полей:

- Оператор;
- Значение.

Перечень доступных операторов и поддерживаемых значений описаны в таблице ниже, **все параметры указываются в скобках [], все строки в кавычках " или ""**:

Оператор	Ожидаемое значение	Доступность в типе данных	Описание	Текст ошибки для полей ввода
=	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <math>\\${valuePath}</math> (пояснения по построению запроса даны в приложении 1)</li> <li>• @now (только для типа данных дата и время)</li> <li>• @today (00:00:00 текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Строка;</li> <li>• Текст</li> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть равно заданному в ограничении значению, с учетом регистра.</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.

		<ul style="list-style-type: none"> <li>• Вычисляемая дата.</li> </ul>		
≠	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <math>\{valuePath\}</math> (пояснения по построению запроса даны в приложении 1)</li> <li>• @now (только для типа данных дата и время)</li> <li>• @today (00:00:00 текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Строка;</li> <li>• Текст</li> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Вычисляемая строка;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть неравно заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM.</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.
<	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <math>\{valuePath\}</math> (пояснения по построению запроса даны в приложении 1)</li> <li>• @now (только для типа данных дата и время)</li> <li>• @today (00:00:00 текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть меньше заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM.</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.

<	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <math>\\${valuePath}</math> (пояснения по построению запроса даны в приложении 1)</li> <li>• @now (только для типа данных дата и время)</li> <li>• @today (00:00:00 текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть меньше заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM.</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.
≤	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <math>\\${valuePath}</math> (пояснения по построению запроса даны в приложении 1)</li> <li>• @now (только для типа данных дата и время)</li> <li>• @today (00:00:00 текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть меньше либо равно заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.

>	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <code>{valuePath}</code> (пояснения по построению запроса даны в приложении 1)</li> <li>• <code>@now</code> (только для типа данных дата и время)</li> <li>• <code>@today (00:00:00</code> текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть больше заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM.</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.
≥	<ul style="list-style-type: none"> <li>• Константа</li> <li>• <code>{valuePath}</code> (пояснения по построению запроса даны в приложении 1)</li> <li>• <code>@now</code> (только для типа данных дата и время)</li> <li>• <code>@today (00:00:00</code> текущего дня) (только для типа данных дата и время)</li> </ul>	<ul style="list-style-type: none"> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата;</li> <li>• Вычисляемое целое число;</li> <li>• Вычисляемое дробное число;</li> <li>• Вычисляемая дата и время;</li> <li>• Вычисляемая дата.</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно быть больше либо равно заданному в ограничении значению</li> <li>• Значение для типа данных “Дата и время” задается на всех локализациях в виде: DD.MM.YYYY HH:MM.</li> <li>• Значение для типа данных “Дата” задается на всех локализациях в виде: DD.MM.YYYY.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Значение должно быть {оператор} {значение ограничения}.
(en) matchRegex  (ru) РЕГВЫРАЖ	<ul style="list-style-type: none"> <li>• Регулярное выражение</li> </ul>	<ul style="list-style-type: none"> <li>• Строка</li> <li>• Текст</li> </ul>	<ul style="list-style-type: none"> <li>• Значение поля должно удовлетворять регулярному выражению заданному в ограничении.</li> <li>• Регулярное выражение задается в формате: <code>^выражение\$</code>.</li> <li>• Пример: ограничение на ввод в поле только цифры от 0 до 9 будет иметь вид: <code>^[0-9]*\$</code>.</li> </ul>	Значение должно удовлетворять регулярному выражению {регулярное выражение}.

(en) Length = (ru) ДЛСТР =	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть равна заданному в ограничении значению.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть = {значение ограничения}
(en) Length ≠ (ru) ДЛСТР ≠	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть неравна заданному в ограничении значению.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть ≠ {значение ограничения}.
(en) Length < (ru) ДЛСТР <	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть меньше заданного в ограничении значения.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть < {значение ограничения}.
(en) Length ≤ (ru) ДЛСТР ≤	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть меньше, либо равна заданному в ограничении значению.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть ≤ {значение ограничения}.
(en) Length > (ru) ДЛСТР >	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть больше заданного в ограничении значения.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть > {значение ограничения}.

(en) Length ≥ (ru) ДЛИСТР ≥	• Числовая константа	• Строка • Текст	<ul style="list-style-type: none"> <li>• Длина значения в поле должна быть больше, либо равна заданному в ограничении значению.</li> <li>• Если активный параметр “множественное значение”, то ограничение применяется к каждому вводимому значению.</li> </ul>	Количество символов должно быть >= {значение ограничения}.
--------------------------------	----------------------	---------------------	---	--

При добавлении ограничения для множественного поля (любого типа) проверка выполняется для каждого элемента массива.

Логика проверки ограничений для массивов:

- Сравнение массива с другим массивом не производится. Система выдает ошибку: Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение}. Значение не может сравниваться с массивом.
- Сравнение массива с “одинарным” значением. Система сравнивает каждый элемент массива с одинарным значением.
- Сравнение “одинарного” значения с массивом не производится. Система выдает ошибку: Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение}. Значение не может сравниваться с массивом.

### Логика работы valuePath в ограничениях

Валидация заданных ограничений по valuePath выполняется по кнопке “Сохранить” при выполнении операции учетчиком. Если валидация по valuePath не выполняется, то система отображает ошибку в зависимости следующих случаев:

1. Система находит учетную единицу, на которую идет ссылка в valuePath, если такой учётной единицы нет, валидация не выполняется, Система выдает ошибку: “Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение valuePath}. Отсутствует учётная единица по пути {valuepath}.”
2. Система находит поле в учётной единице, на которое идет ссылка в valuePath, если такое поле в учётной единице отсутствует, то валидация не выполняется. Система выдает ошибку: “Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение valuePath}. Отсутствует поле {название поля}.”

3. Система сравнивает тип данных в поле учётной единицы, на которое идет ссылка в valuePath с типом данным поля в котором задается ограничение, если тип данных не совпадает, то валидация не выполняется. Система выдает ошибку: “Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение valuePath}. Тип данных не совпадает. Ожидаемый тип: {...}, текущий тип: {...}.”

4. Система находит значение в поле учётной единицы, на которое идет ссылка в valuePath, если значение в поле отсутствует и тип данных не целое число или не дробное число, то валидация не выполняется. Система выдает ошибку: “Ошибка при проверке ограничения в поле {название поля, в котором задано ограничение valuePath}. Отсутствует значение по ссылке в поле {название поля}.”

#### **Отображение ошибок в карточках**

При невыполнении заданных валидационных правил для поля учетной единицы спустя какое-либо время, либо изменения других значений, которые влияют на валидацию значения в поле, в карточках таких учётных единиц отображается ошибка. В таблице карточек в специальной графе отображается символ ошибки для этой карточки (данная графа поддерживает фильтрацию по ошибке и возможность скрыть/отобразить данную графу в таблице). Ячейка таблицы карточек, где находится невалидное значение, отображается красным цветом. При открытии карточки, которая содержит ошибку, в части несоблюдения валидационных правил, поля, которые содержат невалидные значения, содержат маркер ошибки.

#### **Отображение ошибок в справочниках.**

- *Интерфейс настройщика*

При невыполнении заданных валидационных правил для поля в справочниках спустя какое-либо время, либо изменения других значений, которые влияют на валидацию значения в таком поле, в таблице значений такого справочника отображается ошибка. В таблице записей справочника в специальной графе отображается символ ошибки для этой записи (данная графа поддерживает фильтрацию по ошибке и возможность скрыть/отобразить данную графу в таблице). Ячейка таблицы, где находится невалидное значение, отображается красным цветом.

- *Интерфейс учетчика*

На UI справочников, у учетчика графа по отображению ошибки в записи по-умолчанию скрыта. Ячейка содержащая невалидное значение красным не подсвечивается, если скрыта графа с отображением ошибки.

\*ValuePath, который ведет на вычисляемое поле УЕ, работает только с полями этой же УЕ. ValuePath, который ведет по ссылкам (больше одного сегмента) на вычисляемое поле, не работают, так как вычисляемые поля не хранятся в базе данных.

Поддерживаемый синтаксис для конфигурирования отчетов и журналов

Таблица П.4.1. Список поддерживаемых значений в triggerFields

Параметр	Возможные значения	Описание
path	«valuePath»	Триггерное поле в виде записи valuePath, когда триггер срабатывает на каждое изменение поля в valuePath. Количество полей определяет количество уровней в valuePath.  Например: "location.location.location.location"
	“{valuePath}\$»	Триггерное поле в виде записи valuePath, когда необходимый триггер срабатывает только на весь путь valuePath.  Например: "radionuclideComposition.activity\$", триггер срабатывает только на изменения в поле activity.
onlyChanges	true	Триггер срабатывает только на фактическое изменение поля, A -> B. (A -> A, не триггерятся).
	false	Триггер срабатывает на любое изменение поля, A -> B, A -> A.

Таблица П.4.2. Список поддерживаемых значений в fields

Параметр	Возможные значения	Описание
source	instance	Значение указывает на то, что источник данных при формировании значения будет поле из учетной единицы.
	calculated	Значение указывает на то, что источник данных при формировании значения будет формула.
	empty	Значение указывает на то, что при формировании записи в этом поле будет пустая строка. Как правило значение в это поле добавляется после применения постпроцессинга.
	documents	Значение указывает на то, что поле заполняется документами, которые прикреплены к УЕ.

displayName	Пользовательское значение или ключ для перевода	Заданное значение в этом параметре используется для названия графы журнала для задаваемого поля. Если указывается ключ, что расшифровка ключа указывается в файлах ключей для соответствующего языка.
fieldName	Системное название поля из учетной единицы	Значение в поле берется из поля учетной единицы. Используется только если в source указано значение instance. Если в fieldName будет указано поле с типом данных файл, то значение поля будет название прикрепленного файла.
formula	Формула для вычисления значения	Значение поля высчитывается по заданной формуле. Используется только если в source указано значение calculated. Если в formula задан valuePath, который ведет на поле с типом данных файл, то значение поля будет название прикрепленного файла.
name	Пользовательское значение	Уникальное имя, которое идентифицирует поле. Допускается вводить только символы латинского алфавита.
isVerticalHeader (необязательный параметр) Если параметр не указан, то по умолчанию текст отображается горизонтально)	true	Значение указывает на то, что название графы (Header) поля отображается вертикально.
	false	Значение указывает на то, что название графы (Header) поля отображается горизонтально.

Таблица П.4.3. Список поддерживаемых значений в source

Возможные значения	Описание
instance	Значение указывает на то, что используется поле из учетной единицы. Структура описания поля при source: "instance" следующая: <pre>"source": "instance", "fieldName": "системное название поля УЕ, которое необходимо вывести}", "displayname": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"</pre>

calculated	<p>Значение указывает на то, что поле рассчитывается по определенной формуле. Структура описания поля при source: "calculated" следующая:</p> <pre>"source": "calculated", "formula": "формула для вычисления поля" "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"</pre> <p>Список поддерживаемых формул и операторов указан в таблице П.4.4.</p>
empty	<p>Значение указывает на то, что поле будет пустым и заполнение его при необходимости можно осуществить в постпроцессинге (например постпроцессинг с типом rowNumber). Структура описания поля при source: "empty" следующая:</p> <pre>"source": "empty", "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"</pre>
documents	<p>Значение указывает на то, что поле заполняется документами, которые прикреплены к УЕ. Структура описания поля при source: "documents" следующая:</p> <pre>"source": "documents", "format": "{type} - {name} - {date}", - формат отображения каждого документа поддерживаются переменные {type}, {name}, {date} "separator": "; ", - разделитель, если УЕ содержит несколько документов "operationCode": ["23"], - фильтр для отображения документов только из конкретных операций, пустой массив для всех документов означает что будут выводиться документы, только из операции, которая триггернула данную запись "documentType": [123, 275], - фильтр для отображения документов определенного типа "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля",</pre>

	"name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле"
--	--

Таблица П.4.4. Перечень поддерживаемых аргументов в параметре "formula" для "source": "calculated"

Функция	Поддерживаемые типы данных для аргументов функции	Описание
"[\${поле в ссылаемой УЕ}].[название поля в ссылочной УЕ]"]	<ul style="list-style-type: none"> <li>Ссылка</li> </ul>	Вывод значения поля ссылочной УЕ.
"previousValue('\${название поля ссылки}].[название поля в ссылочной УЕ]')" "previousValue('\${название поля}')"	<ul style="list-style-type: none"> <li>Ссылка</li> <li>Обратная ссылка</li> <li>Поле уч.ед.</li> </ul>	Использование предыдущего значения можно указывать путь \$location.location.mass или поле уч.ед \$mass. Будет использовано значение по выбранному пути, которое было на 1мс назад от даты записи.
"join('\${valuePath}', '{разделитель}')" "	<ul style="list-style-type: none"> <li>Ссылка</li> <li>Обратная ссылка</li> <li>Таблица</li> <li>Множественные значения для: <ul style="list-style-type: none"> <li>Строка;</li> <li>Целое число;</li> <li>Дробное число;</li> <li>Дата и время;</li> <li>Дата.</li> </ul> </li> </ul>	Запись значений полученных по заданному пути, через заданный разделитель.  Результатом выполнения функции является строка.
"recursiveDefinitionFind('{название ссылочного поля по которому идет поиск}', '{условие (= или !=)}', {ID типа учётной единицы}, '{поле найденной уч.ед. при истинности условия}', '{значение по умолчанию}', {глубина})"	<ul style="list-style-type: none"> <li>Ссылка</li> <li>Обратная ссылка</li> </ul>	Аргумент, который осуществляет в поле типа ссылка рекурсивный поиск типа учетной единицы по заданному условию, и выводит результат, если результат истина или ложь.
"recursiveJoin('{название ссылочного поля}', '{название поля, значение которого нужно взять по ссылке}', '{разделитель}', {глубина})"	<ul style="list-style-type: none"> <li>Ссылка</li> <li>Обратная ссылка</li> </ul>	Объединение в одну строку, через заданный разделитель, значений найденных с помощью рекурсии по ссылкам.

		<p>Если необходимо указать несколько названий поля, можно перечислить через ' '. Пример: нужно отобразить имя местоположения, при этом имена контейнеров это поле name, а имена ячеек - cellName. recursiveJoin('location', 'name cellName', '; ', '5')</p>
<pre>"previousRecursiveJoin('{название ссылочного поля}', '{название поля, значение котрого нужно взять по ссылке}', '{разделитель}', {глубина})"</pre>	<ul style="list-style-type: none"> <li>• Ссылка</li> <li>• Обратная ссылка</li> </ul>	<p>Запись в одну строку, через заданный разделитель, значений найденных с помощью рекурсии по ссылкам на момент времени 1мс назад от даты записи.  Если необходимо указать несколько названий поля см. recursiveJoin</p>
<pre>"joinArray('\${valuePath}')" "</pre>	<ul style="list-style-type: none"> <li>• Ссылка</li> <li>• Обратная ссылка</li> <li>• Таблица</li> </ul> <p>Множественные значения для:</p> <ul style="list-style-type: none"> <li>• Строка;</li> <li>• Целое число;</li> <li>• Дробное число;</li> <li>• Дата и время;</li> <li>• Дата.</li> </ul>	<p>Массив значений полученных по заданному пути.  Результатом выполнения функции является массив строк.  Отображается в виде строки через разделитель “;”.</p>
<pre>"getOperationInfo('dateTime   code')"</pre>		<p>Функция для использования информации из операции.  Возможные параметры:</p> <ul style="list-style-type: none"> <li>• dateTime - возвращает дату и время когда было произведено данное действие</li> <li>• code - возвращает код операции в которой было произведено данное действие</li> </ul>
<pre>"getEntityDefinitionId('{valuePath}')" "</pre>	<p>Функция позволяет получить EntityDefinitionId для искомого valuePath.</p>	<pre>getEntityDefinitionId('\${name}')</pre> <p>Результат: EntityDefinitionId</p>

	<p>Возвращаемый тип int.</p> <p>В случае если значение не найдено вернется -1.</p> <p>!!!Важно при указании пути не останавливаться на ссылочном поле, а указать какое-нибудь вещественное поле (name, mass, ...).</p>	<p>текущего Entity, если у него присутствует поле name. getEntityDefinitionId('\$location.name')</p> <p>Результат: EntityDefinitionId того объекта который находится по ссылке location у текущего Entity, если у того присутствует поле name. getEntityDefinitionId('\$nonExistingField')</p> <p>Результат: -1.</p>
<p>Также поддерживаются функции и операторы указанные в приложении 2.</p>		

#### Постпроцессинг:

- В постпроцессинге используются только поля журнала, не поля учетной единицы.
- Журнальное поле используется в постпроцессинге по характеристике name.
- В постпроцессинге могут быть использованы любые журнальные поля, независимо от их source.
- Все элементы постпроцессинга выполняются последовательно. В том порядке, в котором они заданы в кастомизации. Независимо от типа постпроцессинга.

Таблица П.4.5. Описание синтаксиса постпроцессинга и набор используемых аргументов

Параметр	Возможные значения	Описание
Тип hideIdentical		
type	hideIdentical	hideIdentical - скрытие идентичной информации между необходимыми колонками. Обычно используется для журнала перемещений, когда нужно скрыть начальную или конечную часть идентичной информации (например графы Откуда - Куда).
direction	<ul style="list-style-type: none"> <li>• start</li> <li>• finish</li> </ul>	<ul style="list-style-type: none"> <li>• start - скрыть информацию в начале</li> </ul>

		<ul style="list-style-type: none"> <li>• finish - скрыть информацию с конца.</li> </ul>	
fieldNames	массив "name" полей, которые участвуют в постпроцессинге	Поля, с характеристикой "name" для которых необходимо скрыть идентичную информацию (Например from и to).	
segmentSeparator	Разделитель	Разделитель, через который записаны данные в полях участвующих в постпроцессинге.	
placeholder	Пользовательское значение	Плейсхолдер, который отображается вместо скрытой идентичной информации.	
Тип filter			
type	filter	filter - отфильтровывает данные в журнале по необходимому значению.	
criteria	operator	<ul style="list-style-type: none"> <li>• equal - равно</li> <li>• notEqual - не равно</li> <li>• gt - больше чем</li> <li>• gte - больше или равно чем</li> <li>• lt - меньше чем</li> <li>• lte - меньше или равно чем</li> <li>• in - входит в список</li> <li>• notIn - не входит в список</li> <li>• isNull - равно null</li> <li>• isNotNull - не равно null</li> </ul>	Фильтр, который выбирает данные с указанным пользовательским значением в поле, у которого задана характеристика "name" для данного постпроцессинга.
	value	Пользовательское значение	Указывается значение, по которому необходимо применить фильтр исходя из значения оператора. Допустимо указывать массив значений только, если указан оператор in или notIn.
Тип sort			
type	sort	Сортирует данные в журнале по необходимому полю.	
fieldNames	"name": "{name поля для сортировки}", "direction": "asc"	Сортировка значений от меньшего к большему (А-Я) в поле, которое задано в характеристике "name"	
	"name": "{name поля для сортировки}", "direction": "desc"	Сортировка значений от большего к меньшему (Я-А) в поле, которое задано в характеристике "name"	

Тип join		
type	join	заполнение данных в ячейки “исходных“ строк журнала, данными из “конечной” строки.
sourceInstanceId	“name” журнального поля отличное от значение “name” destinationInstanceIds	Поле журнала для исходной УЕ, по которому происходит связь между исходными и конечной строками журнала.  В поле должно храниться единичное значение.
destinationInstanceIds	“name” журнального поля отличное от значение “name” sourceInstanceId	Поле журнала для конечной УЕ, по которому происходит связь между исходными и конечной строками журнала.  В поле может храниться как единичное значение, так и массив.
joiningFields	массив “name“ журнальных полей	Журнальные поля, значения которых будут скопированы из конечной УЕ в исходные УЕ.
merge	true/false	Флаг, будут ли сжаты ячейки (по вертикали) для исходных УЕ.  Мерж выполняется только тогда, когда исходные строки располагаются друг за другом.
hideDestinationLine	true/false	Флаг, будет ли скрыта конечная строка из журнала.
overwrite	true/false	Флаг, будет ли скопировано значение из конечной УЕ в исходную в случае, когда ячейка журнала уже заполнена для исходной УЕ.
Тип rowNumber		
type	rowNumber	устанавливает номер строк в выбранное поле.
fieldName	name поля, в которое будет установлен номер строки	Поле, в которое будет установлен номер строки.





## Теги ClosedXML

Тег	Описание
AddFormulaTag	<p>Тег для заполнения столбца формулой excel.</p> <p><b>Параметры</b></p> <p>DeleteNextRow — логическая переменная.</p> <p>DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,</p> <p>DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: <b>**AddFormulaTag{tagId}End**</b></p> <p>DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.</p> <p>DeleteRowWithTag=true — строка с тегом будет удалена,</p> <p>DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).</p> <p>Formula — формула excel, которую необходимо вставить в столбцы (см. пример ниже).</p> <p>HeaderKeyWord — заголовок (или его часть) колонки, которую нужно запомнить. Необходим для определения ячейки, с которой начнёт работать тег. Значение должно быть уникальным для всей колонки excel.</p>

NumberOfRowsToHeader — тег для случая, когда между тегом и заголовком больше 1 строки. По умолчанию =1.

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

### Пример 1

1. Строка с заголовком колонки:  $A_i/C_{i,5}$
2. Строка с будущим значением (пустая):
3. Строка с тегом: <<AddFormula  
Formula=["INDIRECT(ADDRESS(ROW(),3))/INDIRECT(ADDRESS(ROW(),4))"] DeleteNextRow=true .  
HeaderKeyword=["Ai/Ci,5"]>>  
4. Строка с флагом окончания действия тега: \*\*AddFormulaTag1End\*\*

	$A_i/C_{i,5}$
<<AddFormula Formula=["INDIRECT(ADDRESS(ROW(),3))/INDIRECT(ADDRESS(ROW(),4))"] DeleteNextRow=true HeaderKeyword=["Ai/Ci,5"]>> **AddFormulaTag1End**	

### Результат

$A_i^*$ , Бк/г	$C_{i,5}^{**}$	$A_i/C_{i,5}$
137	0.5	274
60	0.1	600
90	2	45

илионхлила в отходах Бк/г

## Пример 2

```
<<AddFormula Formula=["IF(INDIRECT(ADDRESS(ROW(),4))<1,\"Упаковка PAO относится к ОНАО\", \"Упаковка PAO не относится к ОНАО\")"] DeleteNextRow=true HeaderKeyword=["Заключение"] NumberOfRowsToHeader=5 DeleteRowWithTag=true>>
```

16	№	Радионуклид	Удельная активность (A <sub>i</sub> ), Бк/г	Σ <sub>i</sub> A <sub>i</sub> /C <sub>i</sub> 4*	Критерий отнесения к ОНАО	Заключение
17		{{string.Join(";", Item.Fields.Where(i =>	{{string.Join(";", Item.Fields.Where(i =>			
18						
19	Deleterator=[";"] NumberOfAdditionalColumns="1">>				Σ <sub>i</sub> A <sub>i</sub> /C <sub>i</sub> 3 < 1	
20	RowsToHeader=5 DeleteRowWithTag=true					
21	Суммарная активность: {{Item.Fields.Where(i => i.Name == "radionuclideComposition").First().Entities.SelectMany(i => i.ZipAndAddFormula)}					
22						<<AddFormula Formula=["IF(INDIRECT(ADDRESS(ROW(),4))<1,\"Упаковка PAO относится к ОНАО\", \"Упаковка PAO не относится к ОНАО\")"] DeleteNextRow=true HeaderKeyword=["Заключение"] NumberOfRowsToHeader=5 DeleteRowWithTag=true>>
23						**AddFormulaTag1End**

**ClearCellsContentTag** Очищает ячейки колонки.

**DistinctTag**

Тег, который берёт строку в ячейке над ячейкой с тегом, разбивает эту строку в массив, удаляет из него повторяющиеся значения, снова собирает в строку и пишет в ту же ячейку.

**DeleteRowWithTag** — логическая переменная, нужна в случае, когда **ClosedXML** не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

**DeleteRowWithTag=true** — строка с тегом будет удалена,

**DeleteRowWithTag=false** — строка с тегом не будет удалена. (По умолчанию **DeleteRowWithTag=false**).

	<p>Separator — разделитель, по которому строка разбивается на массив, и по которому элементы массива опять собираются в строку.</p>
FilterFieldTag	<p>Тег для фильтрации строк, работает по ячейкам колонки. Если содержимое ячейки удовлетворяет указанному условию, строка остаётся, если нет — удаляется.</p> <p><b>Параметры</b></p> <p>Contains</p> <p>NotContains</p> <p><b>Примеры</b></p> <p>&lt;&lt;FilterField Contains=["TRUE"]&gt;&gt; (удаляются строки, у которых значение ячейки столбца не равно true)</p> <p>&lt;&lt;FilterField NotContains=[""]&gt;&gt; (удаляются строки с пустыми ячейками в столбце)</p>
NumberRowsTag	<p>Тег для нумерации строк таблицы.</p> <p><b>Параметры</b></p> <p>DeleteNextRow — логическая переменная.</p> <p>DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,</p> <p>DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: <b>**NumberRowsTag{tagId}End**</b></p> <p>DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого</p>

поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

DeleteRowWithTag=true — строка с тегом будет удалена,

DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

В первую ячейку, которая должна быть пронумерована, нужно написать **\*\*NumberRowsTag{tagId}Start\*\***

### Примеры

№ п/п
<b>**NumberRowsTag1Start**</b>
<<NumberRows>>
<b>**NumberRowsTag1End**</b>

### SetToLowerItemsTag

Тег, который идёт по столбцу сверху вниз и перебирает значения ячеек.

Если значение ячейки не равно значению "**\*\$^\$#**", оно [исходное значение] запоминается, а ячейке присваивается "**\*\$^\$#**".

Если значение ячейки равно "**\*\$^\$#**", то ей присваивается последнее запомненное значение, отличное от "**\*\$^\$#**".

### Примеры

Место хранения	Дата проверки	Результат проверки
	<code>{{Convert.ToDateTime(Item.FieldValues.Where(i =&gt; i.Definition.Name ==</code>	<code>{{Item.FieldValues.Where(i =&gt; i.Definition.Name == "inventoryResult").First().Value}}</code>
EntityValue.Field	<code>*\$%^\$%/#</code>	<code>*\$%^\$%/#</code>
<<ColsFit>>	<<FilterField NotContains="">><<SetToLowerItems>>	<<SetToLowerItems>><<LowFilterField NotContains=""*\$%^\$%/#">>

## SplitToRowsTag

Тег, который разбивает строку на несколько по указанному разделителю.

### Параметры

DeleteNextRow — логическая переменная.

DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,

DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).

Флаг окончания действия тега: `**SplitToRowsTag{tagId}End**`

NumberOfAdditionalColumns — количество колонок справа, значения которых также должны быть разделены на строки по тому же разделителю. Если ячейки слитые, они не считаются за одну (каждая считается отдельно).

Separator — массив разделителей.

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

### Примеры

1. Строка с заголовком колонки: Наименование СИ

2. Строка с будущим значением: `{{string.Join(";", Item.Fields.Where(i => i.Name == "measuringInstruments").First().Entities.SelectMany(i => . i.Fields).ToList().Where(i => i.Name == "fullName").Select(i => i.Values).SelectMany(i => i).ToArray())}}`
3. Строка с тегом: `<<SplitToRows Separator="[";"]" NumberOfAdditionalColumns="4" Id="1">>`
4. Строка с флагом окончания действия тега: `**SplitToRowsTag1End**`

	A	B	C	D	E	F	G	H
7	Применяемые средства измерения							
8	Наименование СИ			Заводской номер	Номер свидетельства о поверке	Дата следующей поверки		
9	<code>first().Entities.SelectMany(i =&gt; i.Fields).ToList().Where(i =&gt; i.Name == "measuringInstruments").First().Entities.SelectMany(i =&gt; . i.Fields).ToList().Where(i =&gt; i.Name == "fullName").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>							
10	<code>rows Separator="[";"]" NumberOfAdditionalColumns="4" &lt;&lt;ToDateTime Format="["dd.MM.yyyy</code>							
11	<code>**SplitToRowsTag1End**</code>							
12	Идентифицированные радионуклиды							
13	№	Радионуклид	Измеренная удельная активность, Бк/г	Неопределенность измерения, Бк/г		Удельная активность, Бк/г		
14	<code>RowsTagRadionuclide").Select(uctures.SelectMany(i =&gt; i.Fields).ToList().Where(i =&gt; i.Name == "measuringInstruments").First().Entities.SelectMany(i =&gt; . i.Fields).ToList().Where(i =&gt; i.Name == "fullName").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>							
15	<code>numberR";"]" NumberOfAdditionalColumns="5" Id="2"&gt;&gt;</code>							
16	<code>RowsSplitToRowsTag2End**</code>							

**Результат**

Применяемые средства измерения				
Наименование СИ	Заводской номер	Номер свидетельства о поверке	Дата следующей поверки	
device1si	device1f	device1-number	5/18/2030 2:37:00 PM	
device2si	device2f	device2certificate	5/22/2053 4:53:00 PM	
device1si	device1f	device1-number	5/18/2030 2:37:00 PM	
device2si	device2f	device2certificate	5/22/2053 4:53:00 PM	
Идентифицированные радионуклиды				
№	Радионуклид	Измеренная удельная активность, Бк/г	Неопределенность измерения, Бк/г	Удельная активность, Бк/г
1	Cs-137	5		137
2	Co-60			60
3	Sr-90	1		90
4	Cs-137	111	15	546

<b>ToDateTimeTag</b>	<p>Тег для перевода даты и времени в ячейке из одного формата в другой.</p> <p><b>Параметры</b></p> <p><b>TagId</b> — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.</p> <p><b>DeleteNextRow</b> — логическая переменная.  DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,  DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: <b>**ToDateTimeTag{tagId}End**</b></p> <p><b>Format</b> — требуемый итоговый формат даты.</p> <p><b>Примеры</b></p>
----------------------	--

	<p>&lt;&lt;ToDateTime Format=["dd.MM.yyyy"]"&gt;&gt;</p>
ZipAndAddFormulaTag	<p>Тег, который разделяет строку, полученную из двух массивов, выделяет эти массивы, комбинирует их в один массив, затем добавляет в ячейку формулу с новым массивом.</p> <p><b>Параметры</b></p> <p>DeleteNextRow — логическая переменная.  DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,  DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: <b>**ZipAndAddFormulaTag{tagId}End**</b></p> <p>HeaderKeyWord — заголовок (или его часть) колонки, которую нужно запомнить. Необходим для определения ячейки, с которой начнёт работать тег. Значение должно быть уникальным для всей колонки excel.  TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.  ListSeparator — разделитель массивов в исходной строке.  ItemSeparator — разделитель элементов массивов в исходной строке.  NumberOfRowsToHeader — тег для случая, когда между тегом и заголовком больше 1 строки. По умолчанию =1.  ZipAction — элементы нового массива будут строками, полученными по принципу {i-й элемент 1-го массива} + {ZipAction} + {i-й элемент 2-го массива}.  ListAction — разделитель, через который будут соединены элементы нового массива.</p> <pre>cell.FormulaA1 = string.Join(listAction, firstList.Zip(secondList, (first, second) =&gt; first + zipAction + second));</pre> <p><b>Пример 1</b></p> <p>1. Строка с заголовком колонки: Расчетное значение <math>\sum_i A_i/C_i,5^{***}</math></p>

2. Строка с будущим значением: `{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i => i).ToArray())}}*{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "radionuclide").Select(i => i.Entities).SelectMany(i => i).ToList().SelectMany(i => i.Fields).ToList().Where(i => i.Name == "exemptionRateTable5").Select(i => i.Values).SelectMany(i => i).ToArray())}}`
3. Строка с тегом: `<<ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]" ListAction="["+"]" HeaderKeyword="["Расчетное"]">>`
4. Строка с флагом окончания действия тега: `**ZipAndAddFormulaTag1End**`

```

        Расчетное значение  $\sum_i A_i/C_{i,5}^{***}$ 
        {{{string.Join(";", Item.Fields.Where(i => i.Name ==
"radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i =>
i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i =>
i).ToArray())}}*{{string.Join(";", Item.Fields.Where(i => i.Name ==
"radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i =>
i.Name == "radionuclide").Select(i => i.Entities).SelectMany(i => i).ToList().SelectMany(i =>
i.Fields).ToList().Where(i => i.Name == "exemptionRateTable5").Select(i =>
i.Values).SelectMany(i => i).ToArray())}}}
<<ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]"
ListAction="["+"]" HeaderKeyword="["Расчетное"]">>
**ZipAndAddFormulaTag1End**

```

**Результат**

Расчетное значение $\sum_i A_i/C_{i,5}^{***}$
<b>919</b>

## Пример 2

<<ZipAndAddFormula ListSeparator="["\*"]" ItemSeparator="[";"]" ZipAction="["/"]" ListAction="["+""]" HeaderKeyword="["Σi"]" NumberOfRowsToHeader=5 >

№	Радионуклид	Удельная активность (Ai), Бк/г	$\Sigma_i A_i/C_i$ *	Критерий отнесения к ОНАО	Заключение
8	<code>{{string.Join(";", Item.Fields.Where(i =&gt; i.Name == "radionuclideComposition").First().Entities.SelectMany(i =&gt; i.Fields).ToList()).Where(i =&gt; i.Name == "specificActivityOfEachRadionuclide").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>	<code>{{string.Join(";", Item.Fields.Where(i =&gt; i.Name == "radionuclideComposition").First().Entities.SelectMany(i =&gt; i.Fields).ToList()).Where(i =&gt; i.Name == "specificActivityOfEachRadionuclide").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>	<code>{{string.Join(";", Item.Fields.Where(i =&gt; i.Name == "radionuclideComposition").First().Entities.SelectMany(i =&gt; i.Fields).ToList()).Where(i =&gt; i.Name == "specificActivityOfEachRadionuclide").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>	$\Sigma_i A_i/C_i < 1$	
1	Суммарная активность: <code>{{Item.Fields.Where(i =&gt; i.Name == "radionuclideComposition").First().Entities.SelectMany(i =&gt; i.Fields).ToList()).Where(i =&gt; i.Name == "specificActivityOfEachRadionuclide").Select(i =&gt; i.Values).SelectMany(i =&gt; i).ToArray())}}</code>				
2			<code>&lt;&lt;ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]" ListAction="["+""]" HeaderKeyword="["Σi"]" NumberOfRowsToHeader=5 &gt;</code>		<code>&lt;&lt;AddFormula Formula="["IF(INDIRECT(ADDRESS(ROW(),4))&lt;1,"Упаковка PAO относится к ОНАО","Упаковка PAO не относится к ОНАО!")]". DeleteNextRow=true</code>
3			<code>**ZipAndAddFormulaTag1End**</code>		<code>**AddFormulaTag1End**</code>
4	Расчетная дата распада до уровня освобождения от контроля <sup>5</sup> : <code>{{Item.Fields.Where(i =&gt; i.Name == "dateExemptionFromControl").First().Values.Any() ? Convert.ToDateTime(Convert.ToStr</code>				

## ZipTag

Тег, который разделяет строку, полученную из двух или более массивов, выделяет эти массивы, комбинирует их в один массив, затем добавляет в ячейку строку, полученную из этого нового массива. Тег действует на 1 ячейку, расположенную над ячейкой с тегом.

ListSeparator — разделитель массивов в исходной строке.

ItemSeparator — разделитель элементов массивов в исходной строке.

ZipAction — элементы нового массива будут строками, полученными по принципу {i-й элемент 1-го массива} + {ZipAction} + {i-й элемент 2-го массива}.

ListAction — разделитель, через который будут соединены элементы нового массива.

DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

DeleteRowWithTag=true — строка с тегом будет удалена,

DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).

NumberOfLists — количество массивов в строке. По умолчанию =2.

## Пример

Источник образования	<code>{{string.Join(";", Item.Items.SelectMany(i =&gt; i.Fields).Where(i =&gt; i.Name == "placeOfFormation").SelectMany(i =&gt; i.Entities).ToList()).SelectMany(i =&gt; i.Fields).Where(i =&gt; i.Name == "building").SelectMany(i =&gt; i.Entities).ToList()).SelectMany(i =&gt; i.Fields).Where(i =&gt; i.Name == "name").Select(i =&gt; i.Values.First()).ToArray()}}</code> <code>&lt;&lt;Zip ListSeparator=["***"] ItemSeparator=[";"] ZipAction=["/"] ListAction=[";"] NumberOfLists=3&gt;&gt;&lt;&lt;Distinct Separator=[";"] DeleteRowWithTag=true&gt;&gt;</code>
----------------------	---

Ячейка с будущим значением: `{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "placeOfFormation").SelectMany(i => i.Entities).ToList()).SelectMany(i => i.Fields).Where(i => i.Name == "building").SelectMany(i => i.Entities).ToList()).SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray()}}`  
**`{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "placeOfFormation").SelectMany(i => i.Entities).ToList()).SelectMany(i => i.Fields).Where(i => i.Name == "room").SelectMany(i => i.Entities).ToList()).SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray()}}`**  
`{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "deliveryDepartment").SelectMany(i => i.Entities).ToList()).SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray()}}`

Ячейка с тегом: `<<Zip ListSeparator=["***"] ItemSeparator=[";"] ZipAction=["/"] ListAction=[";"] NumberOfLists=3>><<Distinct Separator=[";"] DeleteRowWithTag=true>>`

## Результат

Источник образования	<b>building1kks/room1kks/d1; building2kks/room2.2kks/d1</b> <small>(здание/пом., организация/подразделение)</small>
----------------------	--

Sort	Сортирует область по столбцу, для которого он указан. Параметры Desc и Asc (Asc – по умолчанию) указывают порядок сортировки. Возможна одновременная сортировка по нескольким столбцам. Сортировка происходит справа налево, то есть сначала сортируется самый правый столбец, затем следующий слева и т.д.
Asc	То же, что и <code>&lt;sort asc&gt;</code>
Desc	То же, что и <code>&lt;sort desc&gt;</code>
Group	Создает промежуточные итоги по столбцам, в которых указаны теги итогов ( <code>&lt;sum&gt;</code> и т.д.), группируя их по столбцу, для которого он указан.

	<p>Предварительно область сортируется по всем столбцам, для которых указаны теги Group, Sort, Desc и Asc. Порядок сортировки для тега &lt;group&gt; указывается дополнительным параметром - Desc или Asc (Asc по умолчанию). В случае, если тег &lt;group&gt; указан для нескольких столбцов, промежуточные итоги группируются по всем этим столбцам. Группировка происходит справа налево, то есть сначала итоги группируются по крайнему справа столбцу, для которого указан тег &lt;group&gt;, затем по столбцу с опцией &lt;group&gt; слева от него и т.д.</p> <p>Тег может использоваться без итоговых функций. В этом случае, данные группируются без промежуточных итогов. Для форматирования строк промежуточных итогов и заголовков групп используется форматирование служебной строки области.</p> <p>Параметр MergeLabels вызывает объединение ячеек группы в группируемом столбце.</p> <p>Параметр PlaceToColumn позволяет указать столбец, в который будет помещен заголовок группы.</p> <p>Параметр DisableSubtotals позволяет отключить создание промежуточных итогов для столбца.</p> <p>Параметр WithHeader позволяет создавать заголовок группы при использовании промежуточных итогов. В случае если обнаружен тег &lt;SummaryAbove&gt;, промежуточные итоги размещает над данными.</p> <p>Параметр TotalLabel служит для задания заголовка для промежуточных итогов, например, "Итого по подразделению".</p> <p>Параметр GrandLabel служит для задания заголовка для промежуточных итогов, например, "Итого".</p>
SummaryAbove	Вспомогательный тег для тега <group>. <SummaryAbove> используется для того, чтобы разместить итоги по группам над данными.
DisableGrandTotal	Запрещает создание всех общих итогов при использовании группировки области с промежуточными итогами.
SUM AVG AVERAGE COUNT COUNTNUMS MAX MIN PRODUCT	Теги итогов. Указание его для столбца вызывает подведение итога по столбцу. Итог вычисляется соответствующей функцией Excel. Итоги размещаются в служебном столбце области. По столбцу можно получить только один вид итога (например, сумму или только среднее). В случае если указано несколько итоговых опций, вычисляется только последний указанный итог.

STDEV VAR	
ColsFit	Вызывает автоматическое выравнивание ширины столбцов по значению в ячейках столбцов всего отчета (если указан для отчета), листа (если указан для листа), колонке всего листа (если указан в первой строке листа), области (если указан для области), колонке области (если указан в строке опций) либо для одной ячейки. В случае если указан для листа и для области на этом листе, вызывается только для листа. Тот же принцип от большего к меньшему применяется и для остальных случаев.
RowsFit	Вызывает автоматическое выравнивание высоты строк по значению в ячейках строк всего отчета (если указан для отчета), листа (если указан для листа), колонке всего листа (если указан в первой строке листа), области (если указан для области), колонке области (если указан в строке опций) либо для одной ячейки. В случае если указан для листа и для области на этом листе, вызывается только для листа. Тот же принцип от большего к меньшему применяется и для остальных случаев.
Delete	Удаляет столбец.

### Описание некоторых контролов в зависимости от типа данных

#### **Локализованная строка**

Локализованная строка — это элемент UI, значение которого может меняться вместе с языком системы, или тип данных, позволяющий создать такой элемент UI.

#### **Принцип работы:**

После ввода и сохранения значения в поле с типом данных “локализованная строка” Система автоматически создает ключ для введенного значения, а само введенное значение является переводом на язык, который активен у пользователя при сохранении. Созданный ключ и значение перевода для него попадает в таблицу на вкладке «Ключи и переводы», которая доступна пользователю с ролью «Настройщик». Добавление перевода на другой язык к созданному ключу возможно выполнить двумя способами:

1) После сохранения формы с введенным значением локализованной строки необходимо перейти на вкладку «Ключи и переводы». В таблице ключей и переводов найти ключ/либо значение перевода, по которому можно идентифицировать поле, инициировать редактирование и затем ввести необходимые значения переводов для каждого из оставшихся языков.

2) После сохранения значения в поле с типом локализованная строка переключить язык, для которого необходимо задать новый перевод. После переключения языка, для которого не задан перевод в локализованной строке, в поле с типом локализованная строка будет отображаться плейсхолдер с переводом по приоритету языков установленным администратором. При внесении нового значения перевода плейсхолдер исчезает, если удалить введенное значение, то плейсхолдер возвращается с переводом по приоритету языков. Например: для настройщика выставлен следующий приоритет языков: Русский, Английский. У настройщика активный русский язык интерфейса, он ввел значение «Привет» в поле локализованная строка и сохранил его. Настройщик переключил язык на английский для которого еще не задан перевод в локализованной строке. В поле локализованная строка Настройщик будет видеть плейсхолдер со значением «(RU) Привет». После нажатия на поле пользователь заполняет значение «Hello» перевода для активного языка, плейсхолдер исчезает. Если пользователь удалит значение «Hello», то в поле будет отображаться «(RU) Привет».

В локализованной строке имеется значок в виде «глобуса» на котором, при наведении указателя, появляется подсказка с ключом и его значениями. При нажатии на значок, происходит копирование в буфер обмена значения ключа.

Например:

Наименование

Отдел ядерной безопасности



Ключ: configuration.catalogs.4.31.field1  
EN:  
RU: Отдел ядерной безопасности  
TR:

### **Исключения в работе локализованной строки.**

При добавлении “Группы” работа локализованной строки отличается. Внесение нового значения в поле “Группа” при добавлении нового поля к типу UE, создает новую группу для полей. Если значение совпадает со значением, которое уже вводилось на текущем языке, новая группа не создается.

На UI для пользователя с ролью учётчик, поля у которых отсутствует перевод на выбранный язык, отображают перевод по приоритету без префиксов.

### **Автозаполняемое поле**

В контроле такого типа имеется возможность рассчитать значение по заданному выражению. (см. Приложение 2)

1) Добавление автоматического рассчитанного значения в поле происходит после нажатия кнопки автозаполнение , которая размещается возле контроля.

2) Пользователь имеет возможность изменить значение, которое получилось в результате расчета, заменив расчетное значение пользовательским. При выполнении такой замены кнопка будет иметь вид .

3) Кнопка автозаполнения значения имеет подсказку в которой отображается две строки:

- Значение автозаполнения – рассчитанное значение;
- Сохраненное значение – предыдущее значение, которое было сохранено.

4) Если при расчете значения произошла ошибка, то кнопка автозаполнения имеет вид , а в подсказке в строке «Значение

автозаполнения» содержит запись: «Не удалось выполнить расчет». При нажатии на кнопку  в контрол, в котором произошла ошибка расчета, добавляется предыдущее сохраненное значение.

### Общий концепт функционирования шаблонов документов

#### Описание настройки (конфигурирования) документов в Системе

Настройка документов в системе делится на две взаимодополняемые части:

- 1) Создание и описание excel шаблона для документа.
- 2) Добавление кастомизации данных для документа.

Первым делом настройщик создает excel шаблон. Затем в шаблоне excel, создается второй лист с именем **customization** на этом листе в ячейке **A1** указывается кастомизация которая будет формировать данные для документа.

Кастомизация содержат следующие данные:

- список полей (поддерживаются source: instance, calculated, empty)
- логика сортировки и фильтрации записей (постпроцессинг).

Первый лист содержит шаблон документа с описанием структуры документа и т.д. и полями, в которые будут подставлены значения, которые описаны в кастомизации.

Шаблон документа содержит следующие данные:

- 1) Дефолтные данные:

- название отчета;
- титульный лист для отчета с дефолтными значениями организации;
- шапка таблицы отчёта;
- другие дефолтные данные (утверждаю, согласовано и т.д.).

2) Указать области данных в шаблоне в соответствии с Приложением 5.

3) Указание полей описанных в манифесте в необходимых ячейках, для формирования таблицы документа в виде `{{item.First(i => i.Name == "name поля из кастомизации").Value}}`;

Применить при необходимости теги описаны в Приложении 6.

#### Процесс реализации логики работы (в системе)

- Найти все учётные единицы, которые должны попасть в документ;
- Отфильтровать и отсортировать УЕ по правилам, заданным в кастомизации (например: активность, лицензируемость);
- Выполнить логику сложных полей;

- Передать данные в шаблон excel и расположить их в необходимых ячейках согласно названиям полей в шаблоне.
- Сформировать документ.

### Структура синтаксиса данных в шаблоне документа

Параметр	Описание
Item	Глобальный объект шаблона, в нем содержатся все данные документа.
DocumentInfo <ul style="list-style-type: none"> <li>• Name</li> <li>• Number</li> <li>• Description</li> <li>• Date</li> </ul>	Информация о документе.
OperationInfo <ul style="list-style-type: none"> <li>• Name</li> <li>• Code</li> </ul>	Информация об операции, в которой документ был сгенерирован.
Items	Массив уч.ед. которые попали в документ, каждая уч.ед. представлена в виде пар ключ значение, где ключ - имя поля из кастомизации, значение - значение этого поля.

### Структура синтаксиса кастомизации документа

Параметр	Описание
fields	Поля, которые входят в документ. Структура описания полей зависит от значений в параметре source и описана в Таблице П.4.2. и П.4.3. Приложения 4
postprocessing	Постпроцессинг - действия с данными после формирования. Как правило это фильтрация, сортировка и другие действия над данными. Структура и поддерживаемые типы постпроцессинга описаны в таблице П.4.5 Приложения 4.

### Пример синтаксиса кастомизации документа

```
{
  "fields": [
    {
      "source": "empty",
      "displayName": "uap.modules.reports-module.fields.rowNumber",
      "name": "rowNumber"
    },
    {
```

```

    "source": "calculated",
    "formula": "[$name]",
    "displayName": "name",
    "name": "name"
  },
  {
    "source": "calculated",
    "formula": "[$mass]",
    "displayName": "currentMass",
    "name": "mass"
  },
  {
    "source": "calculated",
    "formula": "previousValue('$mass')",
    "displayName": "previousMass",
    "name": "prevMass"
  }
],
"postprocessing": [
  {
    "type": "sort",
    "fieldNames": [
      {
        "name": "name",
        "direction": "asc"
      }
    ]
  },
  {
    "type": "rowNumber",
    "fieldName": "rowNumber"
  }
]
}

```

### Сервис по заполнению данных в операции из загруженного файла

#### П.10.1. Назначение сервиса

Сервис предназначен для автоматического заполнения полей учетной единицы в операции на основе данных загруженного пользователем файла из спектрометра. Он позволяет извлекать данные из файлов спектрометра с расширением `.spe`, `.ats` с помощью Python-скриптов и автоматически заполнять соответствующие поля учётной единицы в операции. Сервис также добавляет загруженный файл в список документов операции к необходимой учётной единице. Настройка сервиса осуществляется через кастомизацию операции, что позволяет адаптировать его под конкретные бизнес-процессы.

#### П.10.2. Входные данные

- Файл, загруженный пользователем:
  - Файл с поддерживаемым расширением `.spe` или `.ats`.
  - Содержимое файла, которое будет обработано скриптом-парсером.
- Конфигурация сервиса (настраивается в кастомизации операции):
  - Перечень типов учётных единиц, к которым применяется сервис.
  - Список поддерживаемых расширений файлов в операции.
  - Справочник и название в нем поля в котором хранятся радионуклиды для полноценной работы Python-скриптов.
    - Пути к Python-скриптам (парсерам) для каждого расширения.
- Контекст операции:
  - Поля учетной единицы, которые требуется заполнить (список полей указывается в Python-скрипте, а не в конфигурации сервиса).

#### П.10.3. Логика работы сервиса

##### 1. Загрузка файла:

- Пользователь загружает файл в операцию.
- Сервис проверяет расширение файла на соответствие списку поддерживаемых расширений, настроенному в кастомизации операции.

##### 2. Выбор скрипта-парсера:

- На основе расширения файла сервис вызывает соответствующий Python-скрипт (парсер) указанный в настройках кастомизации:

- `spe2json.py` для файла с расширением `.spe`;
- `ats2json.py` для файла с расширением `.ats`;

##### 3. Парсинг файла:

- Python-скрипт анализирует и извлекает данные из файла и преобразует их в структурированный формат (JSON).
- Внутри Python-скрипта указаны:
  - список полей, которые должны быть заполнены в учетной единице;
  - необходимые поля и справочники для полноценной работы Python-скрипта.

#### 4. Заполнение полей учетной единицы:

- Сервис использует извлеченные данные для автоматического заполнения полей учетной единицы, указанных в Python-скрипте.

#### 5. Добавление файла в документы операции:

- Загруженный пользователем файл добавляется в список документов операции к соответствующей учётной единице.

#### П.10.4. Настройка сервиса

- Сервис настраивается через кастомизацию операции, в которой он используется в соответствии с Разделом 3 (Таблица 3.5).

- Тип сервиса: "file".

- Для сервиса необходим справочник с перечнем радионуклидов.

- Для успешной работы сервиса все значения в "parameters" заполняются строго последовательно и в соответствии с указанным синтаксисом.

#### П.10.5. Описание значений для настройки сервиса

Параметр	Значение	Описание
"systemName"	"pyDocumentParser"	Название сервиса
"type"	"file"	Для пользователя с ролью "Учетчик" будет отображен контрол для загрузки файла
"displayName"	<ul style="list-style-type: none"> <li>• Пользовательское значение (перевод контрола для вызова сервиса на UI не происходит);</li> <li>• Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)</li> </ul>	Отображаемое название контрола для вызова сервиса в операции.
"tooltip"	<ul style="list-style-type: none"> <li>• Пользовательское значение (перевод контрола для вызова сервиса на UI не происходит);</li> <li>• Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)</li> </ul>	Всплывающая подсказка для контрола по вызову сервиса
"entityDefinitionIds"	[ID типа учётных единиц]	ID типов УЕ, для которых необходимо инициировать вызов сервиса в операции.

"fileExtensions"	[ {"extension": ".расширение файла", "parser": "bin/parsers/название Python-скрипта.py"}, ... ]	Параметр, в котором указывается для каждого расширения файла, которые могут быть загружены, путь к необходимому Python-скрипту. Доступны следующие значения: Python-скрипты: • для расширения .spe - "extension": ".spe", "parser": "bin/parsers/spe2json.py"; • для расширения .ats - "extension": ".ats", "parser": "bin/parsers/ats2json.py".
"prerequisites"	{ "entityDefinitionId": ID справочника с радионуклидами, "requiredFields": ["название поля справочника, в котором хранятся радионуклиды"] }	Указывается ID справочника с радионуклидами и поле, в котором хранится название радионуклидов.
"parameters"	[ID типа учётной единицы]	ID учётной единицы с типом данных "Структура", который хранит радионуклидный состав в UE.

### П.10.6. Примеры

```
"services": [
  {
    "systemName": "pyDocumentParser"
    "type": "file",
    "displayName": "Спектрометр",
    "tooltip": "Автоматическое заполнение данных из файла спектрометра",
    "fileExtensions": [".spe"],
    "entityDefinitionIds": [11, 12],
    "parameters": [
      [
        { "extension": ".spe", "parser": "bin/parsers/spe2json.py"},
        { "extension": ".ats", "parser": "bin/parsers/ats2json.py"}
      ],
      [{"entityDefinitionId": 165, "requiredFields": ["name"]}
    ]
  }
]
```

### П.10.7. Обработка альтернативных сценариев

Сценарий	Решение
Загружен файл с неподдерживаемым расширением	Сервис возвращает ошибку с сообщением: «Загружен файл с неподдерживаемым расширением. Пожалуйста, загрузите файл с одним

	из следующих расширений: {перечень расширений}»
Ошибка при работе Python-скрипта	Сервис возвращает ошибку с сообщением: «Ошибка выполнения сервиса»
Отсутствует (или неверно указан в кастомизации) Python-скрипт для обработки загруженного файла с расширением	Сервис возвращает ошибку с сообщением: «Ошибка выполнения сервиса: {Отображаемое название сервиса}».