

Creatio IDE

Исходный код (C#)

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

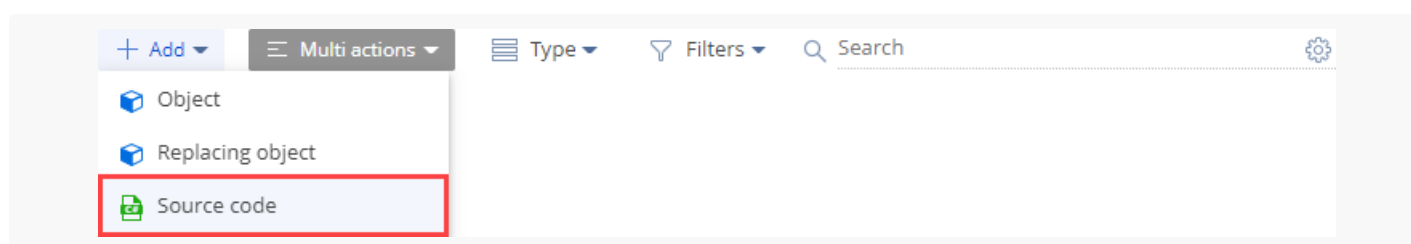
| | |
|------------------------------|----------|
| Исходный код (C#) | 4 |
| Реализовать исходный код | 4 |
| Реализовать замещающий класс | 6 |

Исходный код (C#)

Основы

Конфигурационный элемент типа [Исходный код] ([Source code]) — это сущность, которая реализует бизнес-логику. Позволяет добавлять, удалять и форматировать исходный код добавляемой функциональности на языке C#. **Назначение** конфигурационного элемента типа [Исходный код] ([Source code]) — back-end разработка в приложении Creatio.

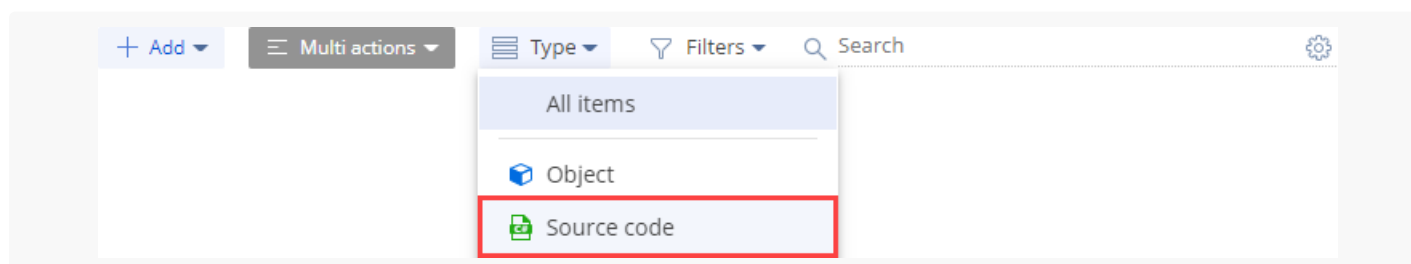
Схема исходного кода, которую позволяет добавить Creatio IDE, представлена значением выпадающего списка [Добавить] ([Add]) панели инструментов рабочей области раздела [Конфигурация] ([Configuration]).



Подробнее о видах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

Конфигурационный элемент типа [Исходный код] ([Source code]) представлен схемой типа [Исходный код] ([Source code]) выпадающего списка [Тип] ([Type]) панели инструментов рабочей области раздела [Конфигурация] ([Configuration]). **Схема** — основа конфигурации Creatio.

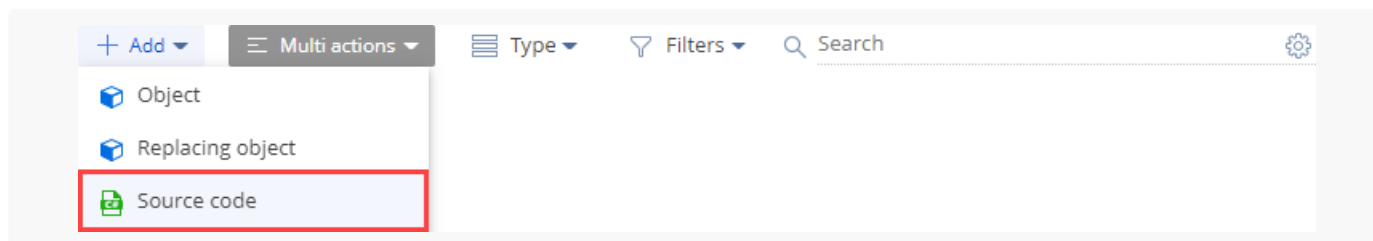
Тип схемы исходного кода представлен на рисунке ниже.



Подробнее о типах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

Реализовать исходный код

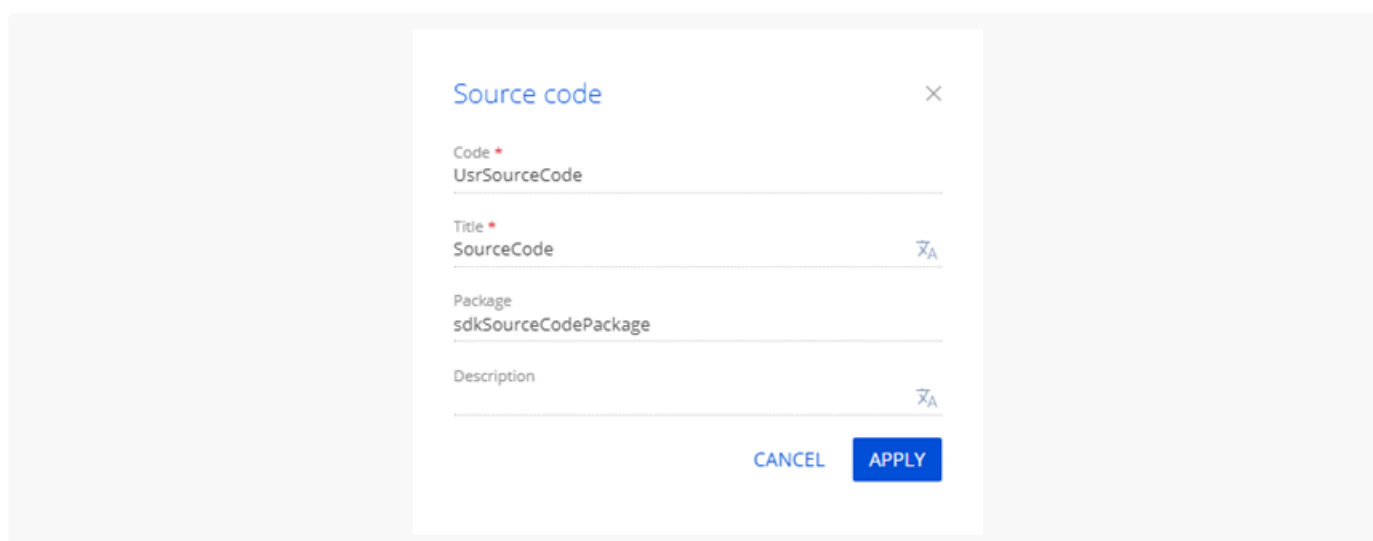
1. [Перейдите в раздел \[Конфигурация \]](#) ([Configuration]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [Добавить] —> [Исходный код] ([Add] —> [Source code]).



3. В дизайнера исходного кода заполните свойства схемы.

Основные свойства схемы:

- [Код] ([Code]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `Usr`), который указан в системной настройке [Префикс названия объекта] ([Prefix for object name], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры.
- [Заголовок] ([Title]) — локализуемый заголовок колонки (обязательное свойство).
- [Пакет] ([Package]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.
- [Описание] ([Description]) — локализуемое описание колонки.



Для применения изменений свойств нажмите [Применить] ([Apply]).

Панель свойств дизайнера исходного кода позволяет:

- Изменить основные свойства схемы (кнопка).
- Задать дополнительные свойства схемы (кнопка).

Дополнительные свойства схемы: [Локализуемые строки] ([Localizable strings]).

4. В дизайнера исходного кода добавьте исходный код. Название класса, который объявлен в исходном коде, должно совпадать с названием схемы (свойство [Код] ([Code]).

Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка или предупреждение). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.

В дизайнера исходного кода Creatio IDE позволяет создать замещающий класс. Для этого

воспользуйтесь инструкцией, которая приведена в пункте [Реализовать замещающий класс](#).

5. На панели инструментов дизайнера исходного кода нажмите [Сохранить] ([Save]) для временного сохранения изменений в метаданных схемы.
6. На панели инструментов дизайнера исходного кода нажмите [Опубликовать] ([Publish]) для выполнения изменений на уровне базы данных.

Реализовать замещающий класс

Принцип замещения классов, в частности создание и использование в конфигурации экземпляров замещаемых классов, имеет свои особенности.

Чтобы **реализовать замещающий класс**:

1. Создайте класс, который является наследником замещаемого класса.
2. Для класса добавьте атрибут `[override]`. Описание атрибута содержится в статье [Атрибут \[Override\]](#).
3. Реализуйте функциональность, которая отличает замещающий класс от замещаемого. Например, реализуйте свойства и методы, которые расширяют функциональность замещаемого класса, перегрузку методов замещаемого класса и т. д.).
 - В замещающем классе для свойств и методов добавьте модификатор `override`.
 - В пользовательском замещаемом классе для свойств и методов, которые необходимо заместить, добавьте модификатор `virtual`.

В базовом классе можно заместить только виртуальные методы или реализовать абстрактные методы. До компиляции все замещающие свойства и методы, которые объявлены без использования ключевого слова `override`, недоступны. Привязка и внедрение зависимостей типов выполняется `op-engine` фреймворком внедрения зависимостей [Ninject](#) только во время выполнения.

Подробнее о замещении конфигурационных элементов читайте в статье [Замещение конфигурационных элементов](#).