

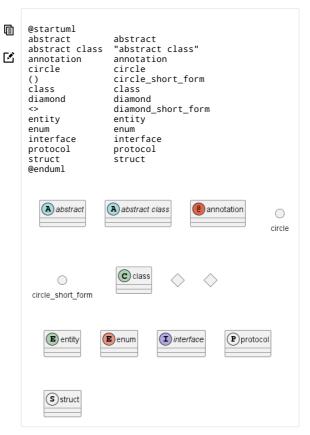


☑ Диаграмма классов

Здесь можно ознакомиться с полезными дополнениями, которые можно использовать в любой из поддерживаемых диаграмм:

- Общие для всех типов диаграмм элементы (комментарии, заметки, управление размером, заголовок и подписи),
- Описание Creole синтаксиса, при помощи которого можно дополнительно форматировать содержимое любых элементов в стиле вики разметки,
- Конфигурирование стилей отображения элементов при помощи управляющей команды skinparam.





[Ссылка для protocol u struct: GH-1028]

[Ref. for protocol and struct: GH-1028, for exception: QA-16258]

WARNING This translation need to be updated. WARNING

👚 В начало

■ Что нового ?

Online Server

≥ Запуск

応 F.A.Q.

У Скачать

Ф Форум

Theme

Препроцессинг

Ш\\ Стандартная библиотека

+ Hitchhiker's Guide

♣ PDF Guide

🔽 🗹 Взаимосвязи между классами

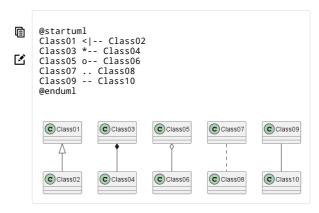
Взаимосвязи между классами (согласно нотации UML) задаются с помощью следующих символов:

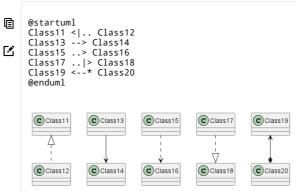
Тип	Символ	Отображение
Наследование	<	\leftarrow
Композиция	*	-
Агрегация	0	├
Зависимость	<	
Ассоциация	<	
Реализация (Имплементация)	<	

Если рассматривать более универсально,

- есть два типа линий:
 - -- рисует сплошную линию
 - .. рисует штриховую линию
- к любой стороне (или сразу к обоим сторонам) линии, соединяющей два элемента, можно добавить разные наконечники:
 - < рисует заостренный наконечник стрелочки
 - <| или ^ рисует наконечник стрелочки в виде треугольника
 - * рисует неконечник стрелочки в виде сплошного ромба
 - о рисует неконечник стрелочки в виде полого ромба
 - # рисует неконечник стрелочки в виде полого квадратика
 - х рисует неконечник стрелочки в виде крестика
 - } рисует неконечник стрелочки в виде обратного треугольника
 - + рисует неконечник стрелочки в виде кружочка с крестиком внутри

Зная эти правила можно нарисовать следующие изображения:





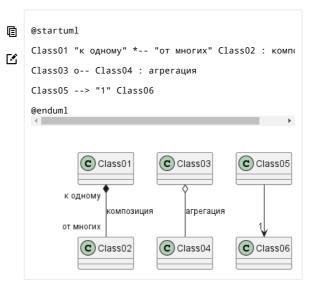




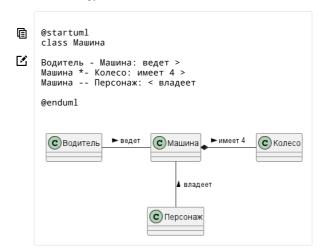


Для каждой связи можно добавить метку. Делается это с помощью добавления символа : после которого указывается текст метки.

Для добавления меток ближе к началу или концу стрелочки можно использовать двойные кавычки "" и помещать метки, обрамленные такими кавычками, перед или после стрелочки как на примере ниже:



Вы можете добавить дополнительные стрелки < или > в начале или в конце метки, чтобы указать на использование одного объекта другим.



последовательност прецедентов классов активности активности в компонент состояний объ В начало

Симвилив

Что нового ?

Если Вы хотите использовать не буквенные (, / , \ , * , # и прочее) или специальные unicode символы в названии класса (а также других элементов), Вам необходимо выполнить следующие условия:

- Поставить кавычки " " вокруг имени элемента которое содержит не буквенные, специальные unicode символы или символы пробела.
- Использовать ключевое слово аз для того, чтобы дополнительно определить короткое ссылочное имя и по нему потом ссылться на этот элемент при определении взаимосвязей



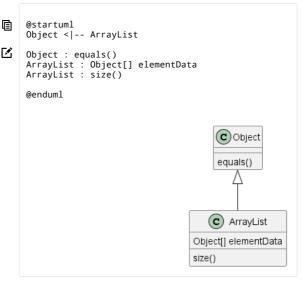
Also note that names starting with ``\$`` are valid, but to assign an alias to such element the name must be put between quotes ``""`.

WARNING This translation need to be updated. WARNING

🔻 🗹 Добавление методов

Для объявления полей и методов вы можете использовать символ : . Для этого каждый раз указываете класс, затем символ : и затем имя поля или метода, который Вы хотите добавить в этот класс.

Система определяет что именно Вы добавили по типу скобок, которые Вы укажете после имени. Для поля данных это должны быть квадратные скобки [], а для метода круглые скобки ().

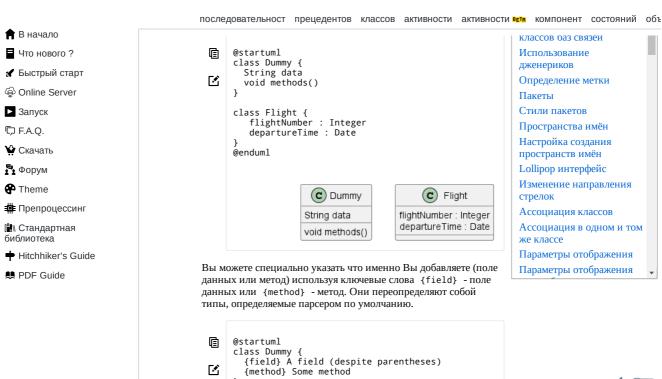


Чтобы не указывать каждый раз имя класса, для которого Вы хотите добавить поле данных или метод, можно все такие поля и методы указать сразу перечислив их внутри фигурных скобок {}.

Декларирующий элемент Взаимосвязи между классами Метки на взаимосвязях Использование не буквенных символов Добавление методов Указание видимости Абстрактные и статические Расширенное тело класса Заметки и шаблоны Больше о заметках Создание заметки для поля данных или метода Заметки на взаимосвязях Абстрактные классы и интерфейсы Скрыть атрибуты, методы... Скрытие классов Удаление классов Hide, Remove or Restore

Online Server

Запуск



⊼ ☑ Указание видимости

@enduml

Определяя методы и поля данных, вы можете использовать символы указания видимости, приведённые в таблице ниже:

© Dummy
A field (despite parentheses)

Some method

Символ	Иконка для поля данных	Иконка для метода	Видимость
-		•	private
#	♦	♦	protected
~	Δ	A	package private
+	0	•	public



Заменить значки их текстовыми представлениями можно используя команду skinparam classAttributeIconSize 0 :







Вы можете определить статические или абстрактные методы и поля данных используя модификаторы {static} и {abstract} соответственно.



Эти модификаторы могут располагаться как в начале, так и в конце строки. Вы так же можете использовать $\{classifier\}$ как замену для $\{static\}$.



🦰 🗹 Расширенное тело класса

По умолчанию, методы и поля автоматически группируются PlantUML. Вы можете использовать разделители, чтобы определить собственный порядок полей и методов. Можно использовать следующие разделители: -- . . . == __ .

Вы также можете использовать заголовки внутри разделителей:







Шаблоны задаются ключевым словом class с парой символов << и >> .

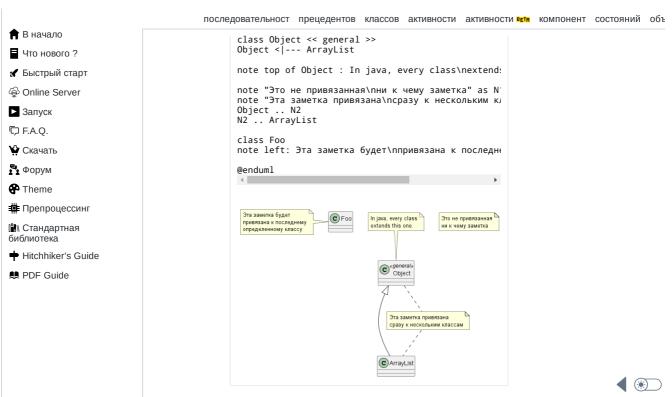
Также вы можете создать и спозиционировать относительно любого класса заметку, используя ключевые слова note left of , note right of , note top of и note bottom of .

Вы также можете создать и спозиционировать заметку относительно последнего определённого класса, используя ключевые слова note left , note right , note top и note bottom .

Ключевым словом note легко создать заметку без привязки и затем привязать её используя символ \dots







⊼ ☑ Больше о заметках

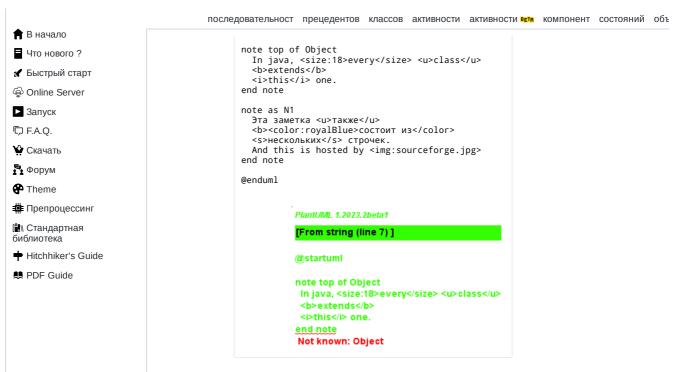
Допускается использование некоторых HTML-тегов (также смотри синтаксис записи Creole), таких как:

-
- <u>>
- <i>
- <s>, , <strike>
- or
- <color:#AAAAAA> or <color:colorName>
- <size:nn> to change font size
- or <img:file> : the file must be
 accessible by the filesystem

Также, заметку можно задавать в несколько строчек. В таком случае, конец заметки определяется по ключевым словам $\,$ end note .





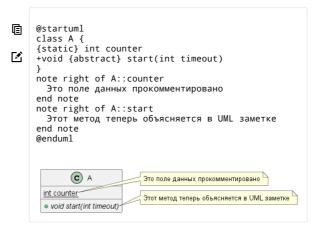


Можно создать и привязать заметку к полю данных или методу.

∆ Ограничения

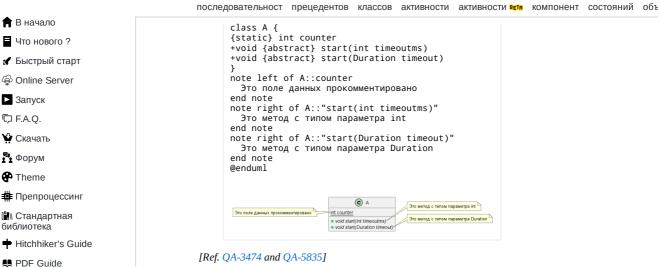
- Нельзя использовать top или bottom (реализованы только left u right)
- Нельзя использовать с разделителем пространства имён ::

Заметка к полю занных или методу



Заметка для разных методов с одинаковым именем





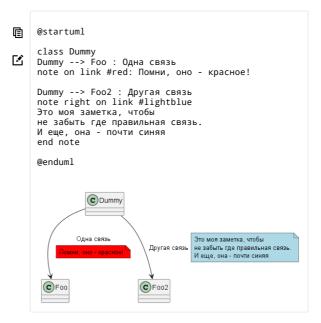
[Ref. QA-3474 and QA-5835]

WARNING This translation need to be updated. WARNING

☑ Заметки на взаимосвязях

Чтобы добавить заметку на связь, сразу после определения такой связи в следующей строке разместите команду note on link.

Вы также можете использовать note left on link, note right on link, note top on link или note bottom on link для того, чтобы добавить заметку и спозиционировать её относительно последней определенной связи.



⊼ ☑ Абстрактные классы и интерфейсы

Вы можете определить класс как абстрактный, используя ключевые слова abstract или abstract class.

Такие классы будут нарисованы курсивом.

A еще Вы можете использовать ключевые слова interface, annotation и enum.



👚 В начало

■ Что нового ?

Online Server

≥ Запуск

Г. F.A.Q.У́Р Скачать№ Форум

Theme

библиотека

Препроцессинг

➡ Hitchhiker's Guide➡ PDF Guide

Ш\\ Стандартная

```
последовательност прецедентов классов активности активности вытивности вытивности вытивности состояний объ
        abstract class AbstractList
        abstract AbstractCollection
        interface List
        interface Collection
        List <|-- AbstractList
Collection <|-- AbstractCollection
        Collection <|- List
AbstractCollection <|- AbstractList
AbstractList <|-- ArrayList</pre>
        class ArrayList {
           Object[] elementData
           size()
        enum TimeUnit {
          DAYS
           HOURS
          MINUTES
        annotation SuppressWarnings
        annotation Annotation {
           annotation with members
          String foo()
String bar()
        @enduml
                                                                                                      ■ TimeUnit
                                                           SuppressWarning:
                                             Annotation
                           A AbstractList
                                          annotation with members
                            C ArrayList
                          Object[] elementDat
```

[Ref. 'Annotation with members' Issue#458]

🔻 🗹 Скрыть атрибуты, методы...

Вы можете параметризовать отображение классов с помощью команды hide/show.

Основная команда: hide empty members . Эта команда скроет атрибуты или методы, если они пусты.

Вместо empty members можно использовать:

- empty fields или empty attributes для пустых полей,
- empty methods для пустых методов,
- fields или attributes, которая скроет поля, даже если они описаны,
- methods которая скроет методы, даже если они описаны,
- members которая скроет поля <u>и</u>методы, даже если они описаны
- circle для обведенного символа перед именем класса,
- stereotype для стереотипа.

Вы также можете указать, сразу после ключевого слова $\,$ hide $\,$ или $\,$ show $\,$:

- class для всех классов,
- interface для всех интерфейсов,
- enum для всех перечислений,
- <<foo1>> для классов, стереотип которых *foo1*,
- существующее имя класса.



⊼ ☑ Скрытие классов

Вы также можете использовать команду show/hide , чтобы скрывать классы.

Это может быть полезно, если Вы подключили большой файл, и хотите скрыть некоторые классы из этого файла после его включения. Сам класс и связи, которые к нему вели будут скрыты, но место, которое он занимал на экране по прежнему будет им занято.



⊼ 🗹 Удаление классов

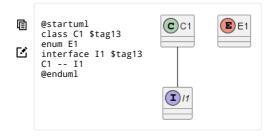
Вы можете использовать команду remove , чтобы удалять классы.

Это может быть полезно, если Вы подключили большой файл, и хотите полностью удалить некоторые классы из этого файла после его включения. В отличии от каманды hide, которая только скрывает класс при отображении его на экране (и при этом этот скрытый класс продолжает занимать на экране место), при использовании команды remove класс и его связи будут



You can put \$tags (using \$) on elements, then remove, hide or restore components either individually or by tags.

By default, all components are displayed:



But you can:

• hide \$tag13 components:



• or remove \$tag13 components:

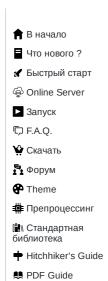


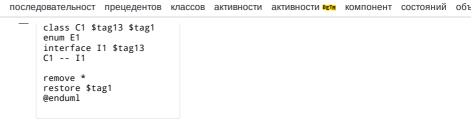
• or remove \$tag13 and restore \$tag1 components:



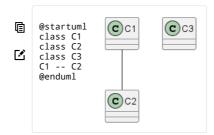
• or remove * and restore \$tag1 components:







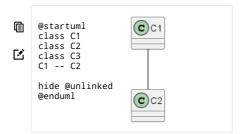
По умолчанию отображаются все классы:



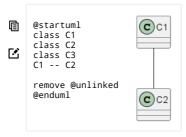


Но вы можете:

• скрывать классы без связей при помощи команды hide @unlinked:



• полностью удалять классы без связей при помощи команды remove @unlinked:



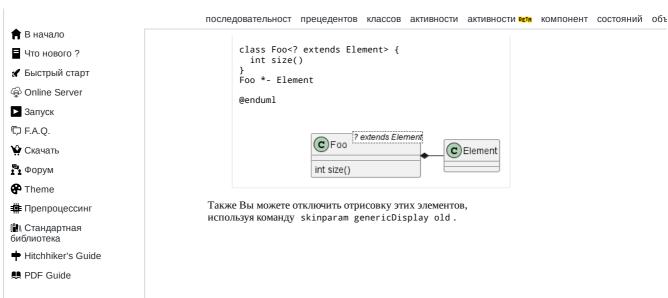
[Adapted from QA-11052]

Использование дженериков

Вы можете использовать угловые скобки < и > чтобы указать на использование дженериков в классе.



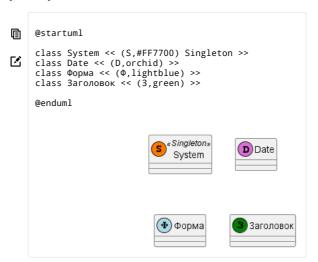
ď



Обычно, метка с буквой (C, I, E or A) испольуется для классов, интерфейсов, перечислений и абстрактных классов.

Но Вы можете создать свою собственную метку для класса при создании шаблона задав для неё собственную букву и цвет (используя html-код для цвета или название цвета).

Русские буквы также можно использовать.



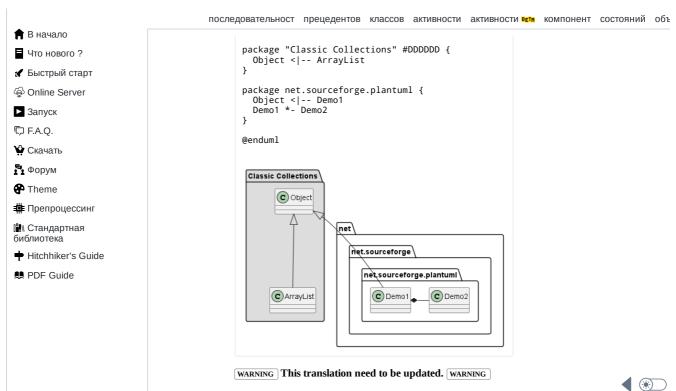
⊼ 🗹 Пакеты

Вы можете создать пакет, используя ключевое слово package , также можно дополнительно указать для него цвет фона (используя html-код для цвета или название цвета).

Обратите внимание, что пакеты могут быть вложенными.



Ľ



\overline 🗹 Стили пакетов

Доступны различные стили для пакетов.

Можно глобально задать стиль по умолчанию с помощью команды: skinparam packageStyle встроенныйСтильПакета, или указать стиль для каждого пакета по отдельности при помощи определения для него шаблона раскаде имяПакета <<встроенныйСтильПакета>>.

На примере ниже можно посмотреть доступные варианты стилей для пакетов:

```
@startuml
scale 750 width
package foo1 <<Node>> {
    class Class1
}

package foo2 <<Rectangle>> {
    class Class2
}

package foo3 <<Folder>> {
    class Class3
}

package foo4 <<Frame>> {
    class Class4
}

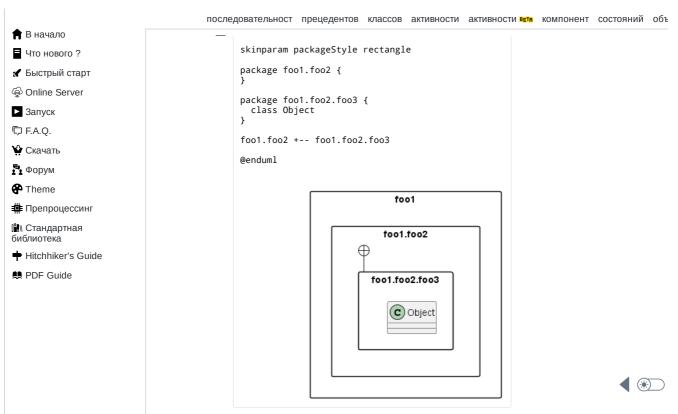
package foo5 <<Cloud>> {
    class Class5
}

package foo6 <<Database>> {
    class Class6
}

@enduml
```

Между пакетами также можно указывать взаимосвязи:





🔼 🗹 Пространства имён

Даже если класс располагается внутри пакета, его имя остается уникальным идентификатором для этого класса. Это значит, что у Вас не может быть двух классов в разных пакетах, но с одинаковым именем.

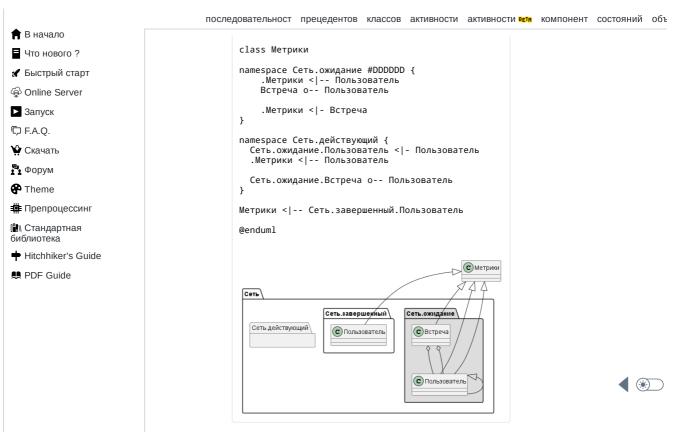
В этом случае, вам следует использовать пространства имен вместо пакетов.

Вы можете ссылаться на классы из других пространств имён по их полному определению. Классы без пространства имён определяются по наличию точки пред их именами.

Обратите внимание, что вы не обязаны явно создавать пространство имен: правильно определенный класс, в имени которого через точку указано пространство имен и затем его имя автоматически попадает в правильное пространство имен.





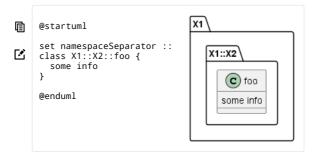


There won't be any difference between namespaces and packages anymore: both keywords are now synonymous.

WARNING This translation need to be updated. **WARNING**

Вы можете задать другой разделитель (не точку) используя команду:

• set namespaceSeparator символыРазделителя.



также Вы можете отключить автоматическое создание пакетов используя команду:

• set namespaceSeparator none.



WARNING This translation need to be updated. **WARNING**

🛊 В начало

■ Что нового ?

Online Server

≥ Запуск

□ F.A.Q.

У Скачать

Форум

Theme

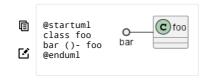
Препроцессинг
 Стандартная библиотека
 → Hitchhiker's Guide
 PDF Guide

последовательност прецедентов классов активности активности ветя компонент состояний объ

🔽 🗹 Lollipop интерфейс

Вы можете задвать интерфейсы в виде "lollipop" для классов, используя следующий синтаксис:

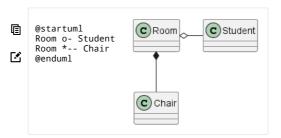
- bar ()- foo
- bar ()-- foo
- foo -() bar



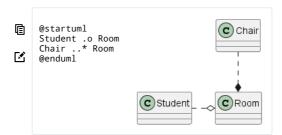
⊼ ☑ Изменение направления стрелок

По умолчанию, связи между элементами задаются при помощи двух тире -- для сплошных линий или при помощи двух точек .. для пунктирных линий и ориентированны вертикально. Можно указать, что связь необходимо изобразить горизонтально, для этого необходимо использовать одиночное тире - для сплошной линии или одну точку . для штриховой линии.



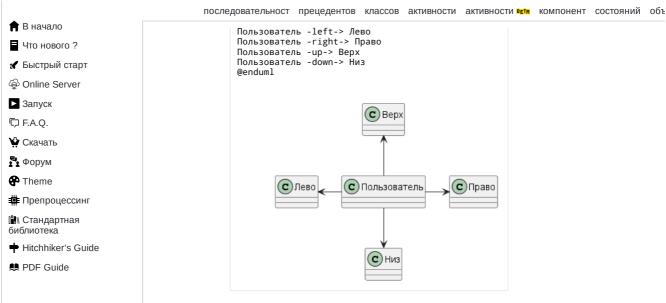


Изменить направление стрелки можно, переместив символ окончания стрелки на другую сторону:



Можно управлять направлением связи добавляя ключевые слова left, right, up или down внутрь стрелки:

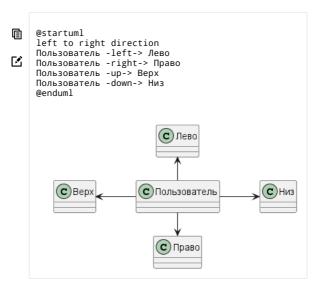




Можно использовать более короткий синтаксис используя только первый символ указанных выше ключевых слов, например -d-или -do- вместо полного слова -down- .

Правда злоупотреблять такой короткой записью не стоит, так как GraphViz обычно дает более хороший результат, когда такая короткая запись HE используется.

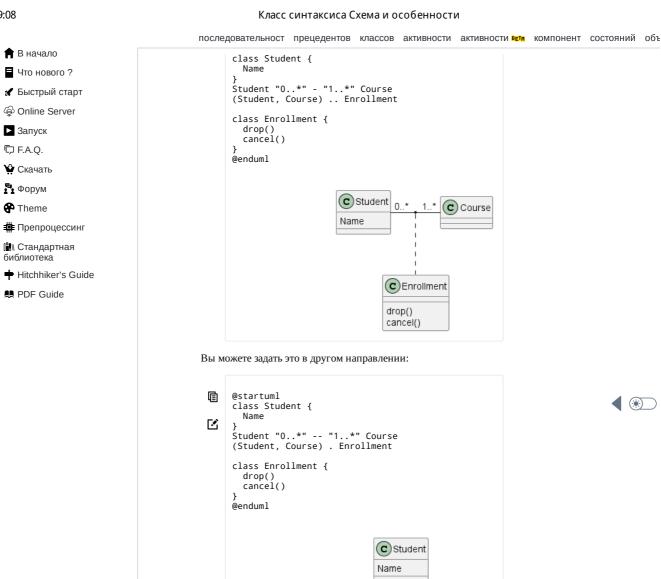
И наконец, можно поменять все направления местами используя ключевое слово left to right direction. Для этого можно сравнить предыдущую диаграмму с диаграммой, представленной ниже:



В Ассоциация классов

Вы можете задать *ассоциацию класса* после того, как была задана связь между двумя классами, как в примере:





⊼ ☑ Ассоциация в одном и том же классе

0...

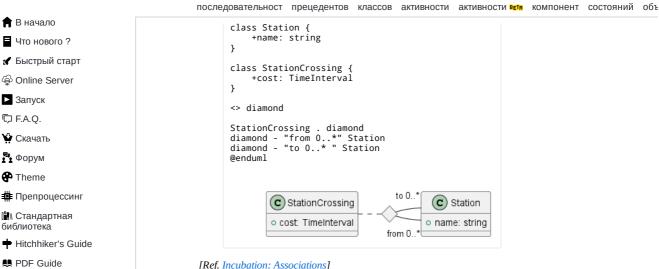
C Course

CEnrollment

drop() cancel()

圓

ď



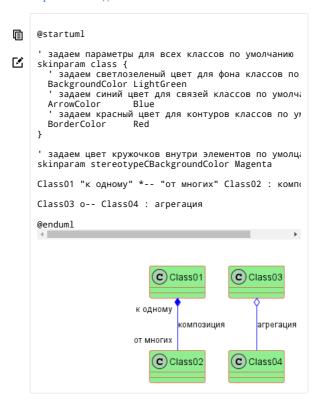
[Ref. Incubation: Associations]

☑ Параметры отображения

Вы можете использовать команду skinparam для глобального изменения шрифтов и цветов различных элементов диаграммы.

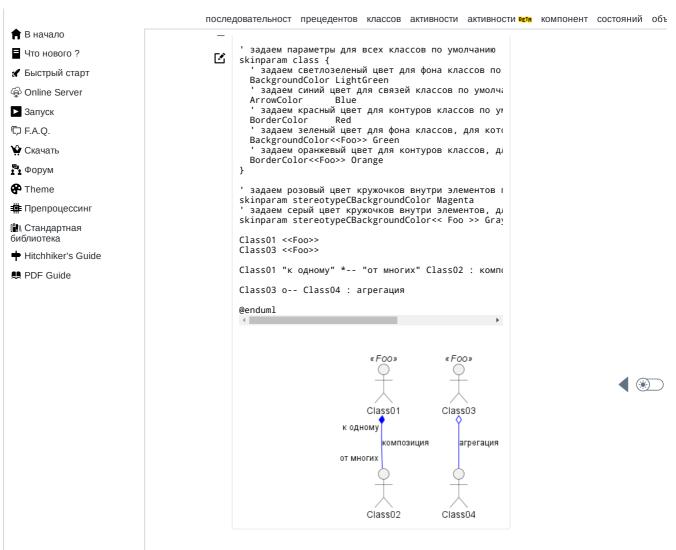
Вы можете использовать данную команду:

- При определении диаграммы, как любую другую команду,
- При подключении файла,
- В конфигурационном файле, указанном в командной строке или в задании ANT.



☑ Параметры отображения для шаблонов

Расширяя пример выше, можно задавать специальные параметры для определенных шаблонов (и применять эти щаблоны затем к тем или иным классам).



⊼ 🗹 Цветовой градиент

Можно объявить индивидуальный цвет для классов или примечаний, используя символ #.

Можно использовать как стандартные названия цветов, так и их RGB-коды в различных вариантах записи.

Так же, возможно задания градиента для фона, задав вместо одного цвета пару цветов и разделив их одним из символов градиента (разные символы отвечают за разные направления градиента):

- |,
- /,
- \,
- -

Для примера:



Ľ



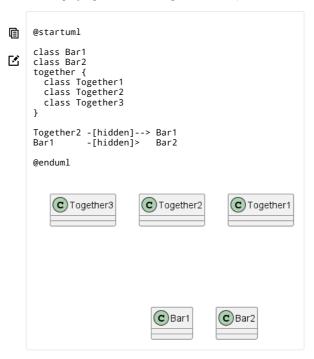


🗸 🗹 Помощь в расположении классов

Иногда диаграмма, формируемая по умолчанию выходит не совсем удачной...

Вы можете объединять несколько классов вместе используя ключевое слово together, тогда рендер постарается расположить объединенные таким образом классы рядом друг с другом так, как если бы они находились в одном пакете.

Для этих же целей можно связать классы между собой связями и скрыть эти связи установив им соответствующее свойство hidden. Тогда рендер также постарается расположить такие связанные классы вместе (причем длинной такой скрытой связи можно регулировать расстояние между классами, а направлением связи можно регулировать взаимное расположение).



☑ Разделение больших файлов

Иногда могут получиться очень большие файлы изображений.

последовательност прецедентов классов активности активности вств компонент состояний объ

в начало

(hpages)x(vpages), где:

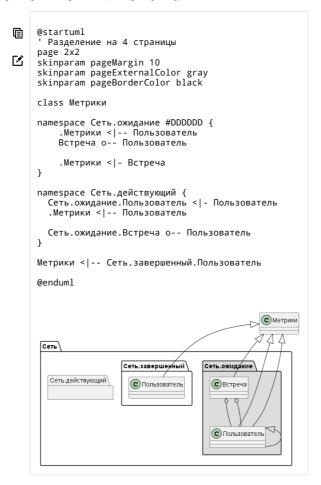
Что нового?

• hpages - это число горизонтальных страниц,

✓ быстрый старт

• vpages - это число вертикальных страниц...

С этой командой также можно использовать команды skinparam параметр значение для настроек цвета разделённых страниц, и размеров их границы (смотри пример).



Вы можете использовать ключевые слова extends и implements для содания связей наследования и реализации (имплементации).

Online Server

- + Hitchhiker's Guide
- 🛤 PDF Guide

библиотека

👚 В начало

■ Что нового ?

Online Server

Запуск

□ F.A.Q.

У Скачать

Форум

Theme

библиотека

♣ PDF Guide

Препроцессинг **Ш**\\ Стандартная

+ Hitchhiker's Guide

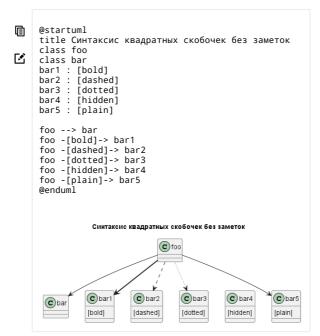
последовательност прецедентов классов активности активности выти компонент состояний объ

или стрелок (синтаксис квадратных

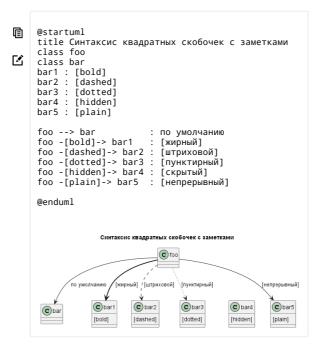
Есть два разных синтаксиса для изменения цвета и стиля связей и стрелок. Возможности этого синтаксиса квадратных скобочек несколько больше, чем возможности встраиваемого синтаксиса (подробнее о нем будет ниже), так как здесь можно полностю

dashed - штриховой, dotted - пунктирный, hidden

• пример для каждого стиля без заметок на связях:



• пример для каждого стиля с заметками на связях:

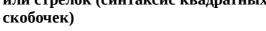


[Adapted from QA-4181]

Цвет линий

Можно задавать цвет для связей:



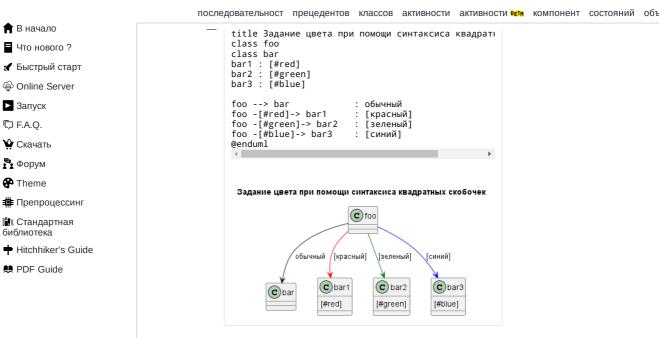


скрывать связи, а также задавать их толщину.

Стиль линий

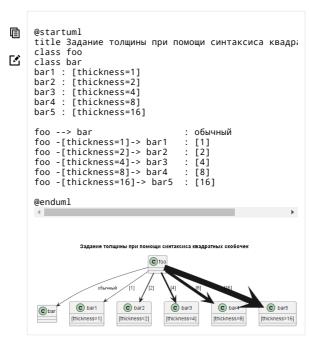
Можно задавать следующие стили для связей: bold - жирный, скрытый, plain - непрерывный:





Толщина линий

Можно задать толщину связей:



[Ref. QA-4949]

Смешивание различных парметров

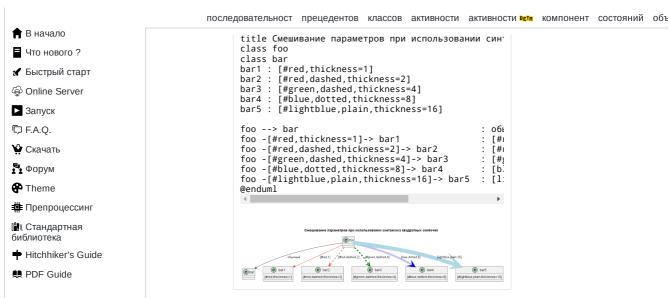


[#1

[#{

Γb.

[1





☑ Изменение цвета и стиля связей или стрелок (встраиваемый синтаксис)

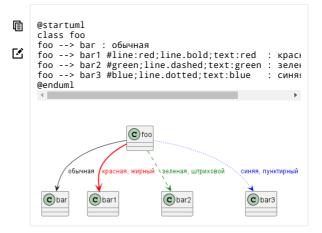
Возможности этого синтаксиса немного меньше, чем синтаксиса квадратных скобочек, который был рассмотрен выше. В частности, здесь нельзя задавать толщину линий, а также нельзя полностью скрывать линии.

Вы можете изменить цвет или стиль индивидуально для каждой связи или стрелочки используя следующий встраиваемый

#color;line.[bold|dashed|dotted];text:color

где:

- # начало режима конфигурирования,
- color цвет стрелки,
- ; разделитель между конфигурационными параметрами,
- line.type тип линии. type может принимать значения: bold - жирная, dashed - штриховая, dotted пунктирная.
- text:color цвет текста заметки на линии.



Пример установки одних и тех же параметров при помощи двух различных синтаксисов можно посмотреть на примере ниже:



👚 В начало

■ Что нового ?

Online Server

Запуск

□ F.A.Q.

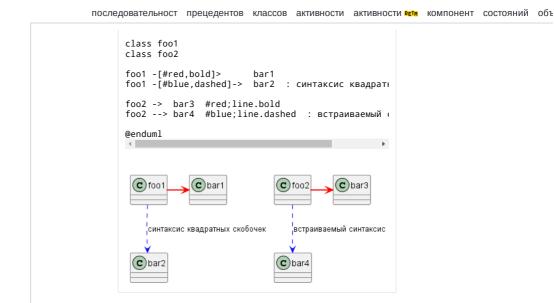
СкачатьФорумТheme

библиотека

♣ PDF Guide

Препроцессинг Препроцессинг

+ Hitchhiker's Guide



[See similar feature on deployment]

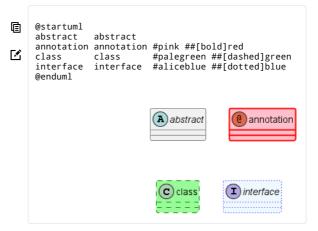


Вы можете изменить цвет или стиль индивидуально для каждой стрелочки используя следующие 2 варианта встраиваемого синтаксиса:

#color ##[style]color

где:

- #color цвет фона элемента,
- ##[style]color стиль отображения и цвет линий элемента.

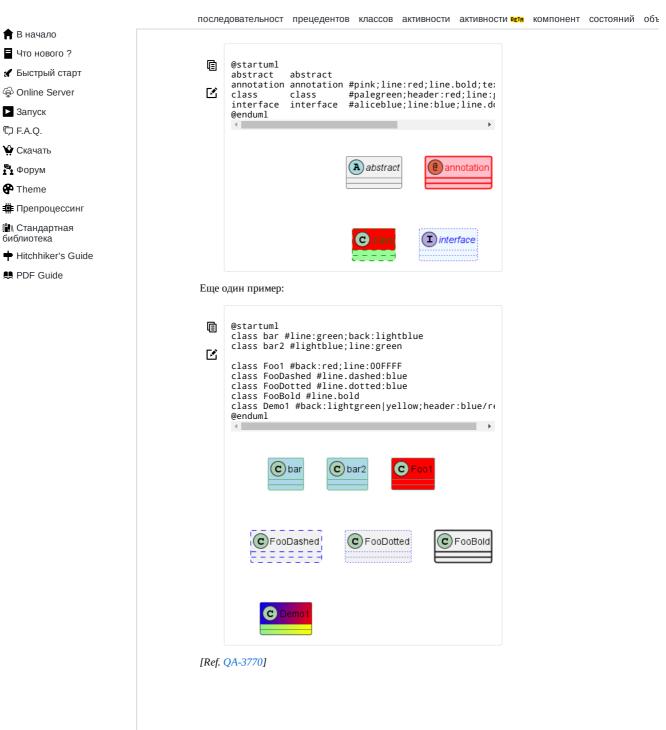


[Ref. QA-1487]

2. #[color|back:color];header:color;line:color;line.
[bold|dashed|dotted];text:color

где:

- # начало режима конфигурирования,
- color или back:color основной цвет фона,
- ; разделитель между конфигурационными параметрами,
- header:color дополнительный цвет фона для заголовка,
- line:color цвет для контуров элемента,
- line.[type] тип линий, из которых состоит контур элемента (type может принимать значения: bold жирный, dashed - штриховой, dotted - пунктирный),
- text:color цвет текста.



⊼ ☑ Связи от/на члены класса

•

👚 В начало

Online Server

ЗапускБ F.A.Q.

У Скачать

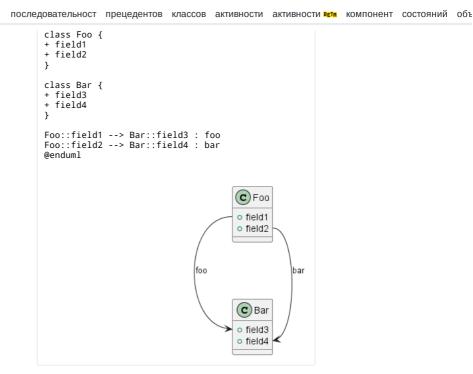
та форум **№** Theme

Препроцессинг

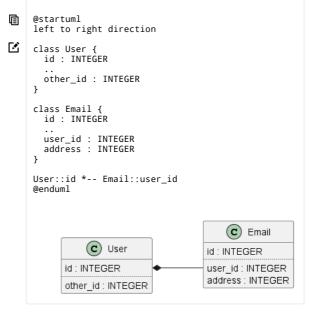
➡ Hitchhiker's Guide➡ PDF Guide

Ш\\ Стандартная

библиотека



[Ref. QA-3636]



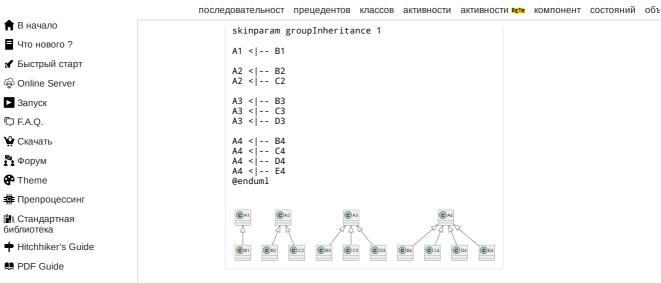
[Ref. QA-5261]

¬ У Группировка стрелок, ведущих к одному элементу

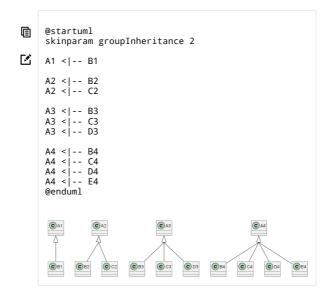
Вы можете включить объединение окончания стрелок, сходящихся к одному элементу, используя команду skinparam groupInheritance X, где X-это минимальное количество стрелок, при котором будет включаться данный режим.

Установка параметра в значение 1 (без группировки)

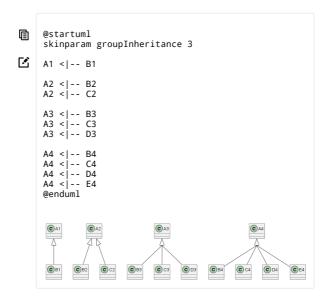
ď



Установка параметра в значение 2 (группировка если 2 и более)



Установка параметра в значение 3 (группировка если 3 и более)



Установка параметра в значение 4 (группировка если 4 и более)



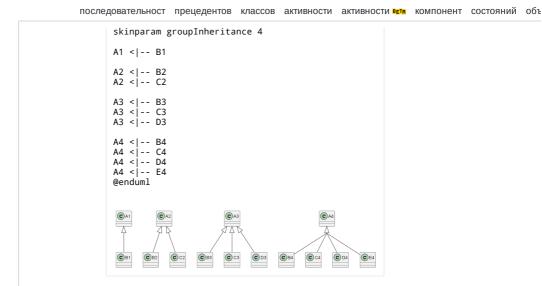
👚 В начало

■ Что нового ?

Online Server

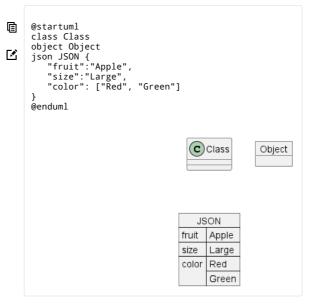
ЗапускБ F.A.Q.

Препроцессинг
 Стандартная библиотека
 → Hitchhiker's Guide
 PDF Guide



[Ссылка на QA-3193, и дефект QA-13532]

Simple example



[Ref. QA-15481]

For another example, see on JSON page.

Privacy Policy Advertise