

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)



## Диаграммы последовательностей

Создание диаграмм последовательностей с помощью PlantUML удивительно просто. Эта простота во многом объясняется удобством синтаксиса, который разработан таким образом, чтобы быть интуитивно понятным и легко запоминающимся.

- **Интуитивно понятный синтаксис:**

Прежде всего, пользователи ценят простой и интуитивно понятный синтаксис PlantUML. Благодаря продуманному дизайну даже новички в области создания диаграмм быстро и без труда постигают его основы.

- **Соотношение текста и графики:**

Еще одной отличительной особенностью является тесное сходство между текстовым и графическим представлением. Благодаря этой гармоничной взаимосвязи текстовые наброски достаточно точно преобразуются в графические диаграммы, обеспечивая целостность и предсказуемость дизайна без неприятных сюрпризов в конечном результате.

- **Эффективный процесс разработки:**

Сильная корреляция между текстом и графическим результатом не только упрощает, но и значительно ускоряет процесс разработки. Пользователи выигрывают от более оптимизированного процесса с меньшими затратами времени на доработку и корректировку.

- **Визуализация в процессе разработки:**

Возможность представить себе конечный графический результат во время работы над текстом - функция, которую многие считают бесценной. Она способствует плавному переходу от первоначального проекта к окончательному варианту, повышая производительность и снижая вероятность ошибок.

- **Простота редактирования и внесения изменений:**

Важно отметить, что редактирование существующих диаграмм не требует особых усилий. Поскольку диаграммы создаются на основе текста, пользователи обнаруживают, что вносить в них изменения гораздо проще и точнее, чем в графические изображения. Все сводится к простому изменению текста, что гораздо проще и менее подвержено ошибкам, чем внесение изменений через графический интерфейс с помощью мыши.

PlantUML обеспечивает простой и удобный подход к созданию и редактированию диаграмм последовательности, удовлетворяя потребности как новичков, так и опытных разработчиков. Он умело использует простоту текстового ввода для создания визуально описательных и точных диаграмм, тем самым зарекомендовав себя как обязательный инструмент в наборе средств создания диаграмм.

Вы можете узнать больше о некоторых [распространенных командах PlantUML](#), чтобы расширить свои возможности по созданию диаграмм.

## Основные примеры

Последовательность -> используется для передачи сообщения между двумя участниками . Участники не обязательно должны быть явно объявлены.

Чтобы иметь пунктирную стрелку, вы используете -->

Также можно использовать <- и <-- . Это не меняет рисунок, но может улучшить читабельность. Обратите внимание, что это справедливо только для диаграмм последовательности, для других диаграмм правила другие.

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

```

Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
@enduml

```



WARNING This translation need to be updated. WARNING

## Объявление участников

При использовании ключевого слова `participant` возможно получить больший контроль над отображением участников.

Порядок перечисления участников задаёт также **порядок отображения** участников по умолчанию.

Использование других ключевых слов (отличных от `participant`) позволяет изменить **форму представления (отображения)** участника:

- `actor`
- `boundary`
- `control`
- `entity`
- `database`
- `collections`
- `queue`

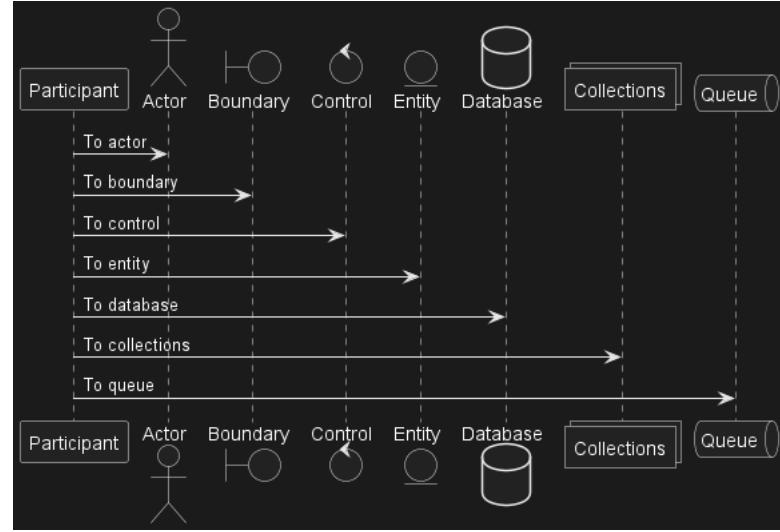
Ос  
Ое  
уч  
Ое  
в  
фс  
Ис  
не  
в  
уч  
Со  
се  
въ  
Из  
ст  
Из  
Ну  
в  
На  
фу  
Ра  
Гр  
со  
Дс  
гр  
Пр  
со  
Др

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```

participant Participant as Foo
actor Actor as Foo1
boundary Boundary as Foo2
control Control as Foo3
entity Entity as Foo4
database Database as Foo5
collections Collections as Foo6
queue Queue as Foo7
Foo -> Foo1 : To actor
Foo -> Foo2 : To boundary
Foo -> Foo3 : To control
Foo -> Foo4 : To entity
Foo -> Foo5 : To database
Foo -> Foo6 : To collections
Foo -> Foo7 : To queue
@enduml

```



Возможно переименовать участника используя ключевое слово `as`.

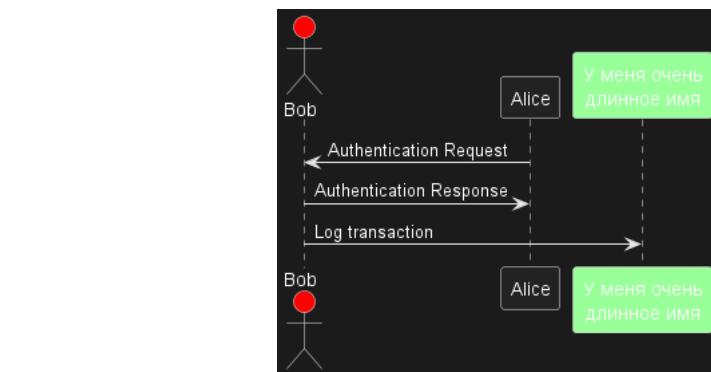
Также возможно изменить цвет фона участника.

```

@startuml
actor Bob #red
' Единственная разница между actor
' и participant - это разная форма отображения
participant Alice
participant "У меня очень\пдлинное имя" as L #99FF99
' Эта строка будет обозначать тоже самое:
    participant L as "У меня очень\пдлинное имя" #99FF99
'

Alice->Bob: Authentication Request
Bob->Alice: Authentication Response
Bob->L: Log transaction
@enduml

```



Возможно изменить порядок следования участников с помощью ключевого слова `order`.

За  
пс  
уч  
въ  
не  
од  
Cr  
Ра  
Сс  
За  
ди  
Пе  
Пр  
Ак  
де  
су  
Re  
О1  
со  
пр  
Бъ  
ак  
де  
со  
Вх  
се

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

[последовательность](#)
[прецедентов](#)
[классов](#)
[активности](#)
[компонент](#)
[состояний](#)


```

participant Last order 30
participant Middle order 20
participant First order 10
@enduml
  
```

## Объявление участника в многострочной форме

Вы можете объявлять участника в многострочной форме.

```

@startuml
participant Participant [
    =Title
    -----
    "SubTitle"
]
participant Bob
Participant -> Bob
@enduml
  
```

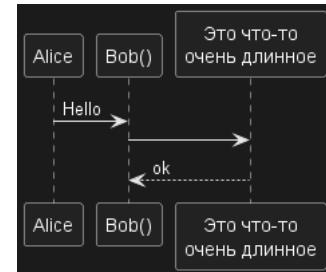
[Ссылка QA-15232]

## Использование небуквенных символов в названиях участников

Вы можете использовать кавычки для задания участников. Также Вы можете использовать ключевое слово `as` для присвоения псевдонимов к этим участникам.

```

@startuml
Alice -> "Bob()" : Hello
"Bob()" -> "Это что-то\очень длинное" as Long
' Эта строчка будет обозначать тоже самое:
' "Bob()" -> Long as "Это что-то\очень длинное"
Long --> "Bob()" : ok
@enduml
  
```



## Сообщения к самому себе

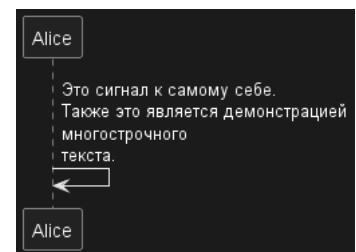
Участник может посыпать сообщения сам себе.

Также возможно создание многострочных сообщений, используя `\n`.

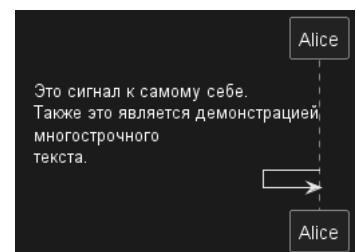
- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```
Alice -> Alice: Это сигнал к самому себе.\nТакже это является демонстрацией  
@enduml
```



```
@startuml  
Alice <- Alice: Это сигнал к самому себе.\nТакже это является демонстрацией  
@enduml
```



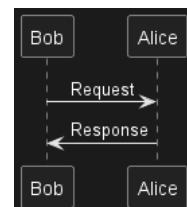
[Ref. QA-1361]

## Выравнивание текста

Расположение текста относительно стрелки можно указывать как `left` (лево), `right` (право) или `center` (центр) используя слово `skinparam sequenceMessageAlign`.

Вы также можете использовать `direction` или `reverseDirection` для выравнивания текста относительно направления стрелки. Больше подробностей и примеров можно найти на странице [skinparam](#).

```
@startuml  
skinparam sequenceMessageAlign right  
Bob -> Alice : Request  
Alice -> Bob : Response  
@enduml
```



## Текст ответного сообщения под стрелкой

Вы можете разместить текст ответного сообщения под стрелкой используя команду `skinparam responseMessageBelowArrow true`.

```
@startuml  
skinparam responseMessageBelowArrow true  
Bob -> Alice : дай пять рублей  
Alice -> Bob : нет, не дам  
@enduml
```



- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

## Изменение стиля стрелок

Можно изменить стиль стрелок следующими способами:

- закончить стрелку с помощью `x` для обозначения потерянного сообщения
- использовать `\` или `/` вместо `<` или `>` для создания только верхней или нижней части стрелки
- повторить окончание стрелки (например, `>>` or `//-`) для тонкой отрисовки
- использовать `--` вместо `-` для создания пунктирной стрелки
- заканчивать символом "`o`" во главе стрелки
- использовать двунаправленные стрелки `<->`

```
@startuml
Bob ->x Alice
Bob -> Alice
Bob ->> Alice
Bob -\ Alice
Bob \\\- Alice
Bob //-- Alice

Bob ->o Alice
Bob o\\-- Alice

Bob <-> Alice
Bob <->o Alice
@enduml
```

## Изменить цвет стрелок

Вы можете изменить цвет отдельных стрелок, используя следующие правила:

```
@startuml
Bob -[#red]> Alice : hello
Alice -[#0000FF]->Bob : ok
@enduml
```

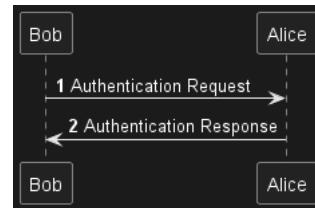
## Нумерация сообщений в последовательностях

Ключевое слово `autonumber` используется для автоматической нумерации сообщений.

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

[последовательность](#)
[прецедентов](#)
[классов](#)
[активности](#)
[компонент](#)
[состояний](#)

```
autonumber
Bob -> Alice : Authentication Request
Bob <- Alice : Authentication Response
@enduml
```



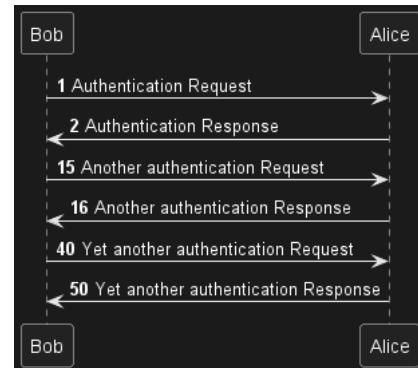
Вы можете обозначить число с которого начнется отсчет autonumber `<start>`, а также число которое будет использоваться в качестве инкремента `autonumber <start> <increment>`.

```
@startuml
autonumber
Bob -> Alice : Authentication Request
Bob <- Alice : Authentication Response

autonumber 15
Bob -> Alice : Another authentication Request
Bob <- Alice : Another authentication Response

autonumber 40 10
Bob -> Alice : Yet another authentication Request
Bob <- Alice : Yet another authentication Response

@enduml
```



Можно задавать формат чисел, указав его в двойных кавычках.

Форматирование выполнено с использованием класса Java `DecimalFormat` (0 означает цифру, # означает цифру или ноль если пусто).

При формировании также можно использовать теги html.

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```

autonumber "<b>[000]</b>" 
Bob -> Alice : Authentication Request
Bob <- Alice : Authentication Response

autonumber 15 "<b>(<u>##</u>)" 
Bob -> Alice : Another authentication Request
Bob <- Alice : Another authentication Response

autonumber 40 10 "<font color=red><b>Message 0 </b></font>" 
Bob -> Alice : Yet another authentication Request
Bob <- Alice : Yet another authentication Response

@enduml

```



Вы также можете использовать autonumber stop и autonumber resume <increment> <format> чтобы соответственно остановить и продолжить автоматическое нумерование.

```

@startuml
autonumber 10 10 "<b>[000]</b>" 
Bob -> Alice : Authentication Request
Bob <- Alice : Authentication Response

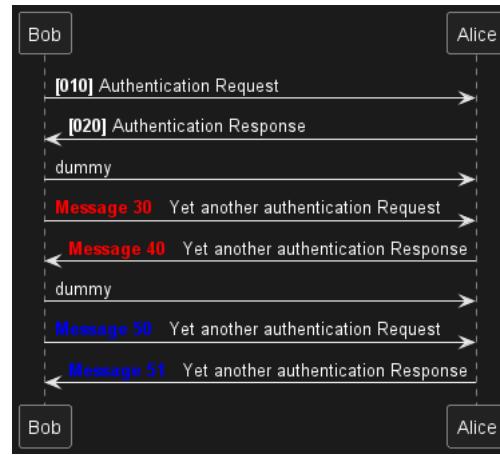
autonumber stop
Bob -> Alice : dummy

autonumber resume "<font color=red><b>Message 0 </b></font>" 
Bob -> Alice : Yet another authentication Request
Bob <- Alice : Yet another authentication Response

autonumber stop
Bob -> Alice : dummy

autonumber resume 1 "<font color=blue><b>Message 0 </b></font>" 
Bob -> Alice : Yet another authentication Request
Bob <- Alice : Yet another authentication Response
@enduml

```



Ваше число, с которого начнется отсчет может быть последовательностью из двух или трех цифр, разделенных знаками . , ; , : . Также это могут одновременно два разных знака из этого списка. Например: 1.1.1 или 1.1:1 .

- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

Для того, чтобы инкрементировать первую цифру последовательности, укажите autonumber inc A . Для инкрементирования второй цифры последовательности укажите соответственно autonumber inc B .

```
@startuml
autonumber 1.1.1
Alice -> Bob: Authentication request
Bob --> Alice: Response

autonumber inc A
'Now we have 2.1.1
Alice -> Bob: Another authentication request
Bob --> Alice: Response

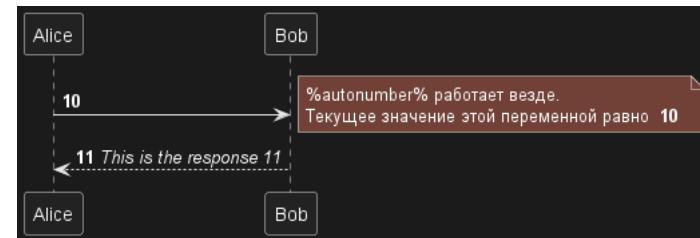
autonumber inc B
'Now we have 2.2.1
Alice -> Bob: Another authentication request
Bob --> Alice: Response

autonumber inc A
'Now we have 3.1.1
Alice -> Bob: Another authentication request
autonumber inc B
'Now we have 3.2.1
Bob --> Alice: Response
@enduml
```



Вы также можете использовать значение autonumber ВМЕСТЕ с %autonumber% переменной:

```
@startuml
autonumber 10
Alice -> Bob
note right
    <U+0025>autonumber<U+0025> работает везде.
    Текущее значение этой переменной равно ** %autonumber% **
end note
Bob --> Alice: //This is the response %autonumber%//
@enduml
```



[Ref. QA-7119]

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность    прецедентов    классов    активности    компонент    состояний

## Название, Заголовок и футтер страницы

Ключевое слово `title` используется, чтобы добавить название страницы.

Также страницы могут иметь заголовок и футтер с помощью ключевых слов `header` и `footer`.



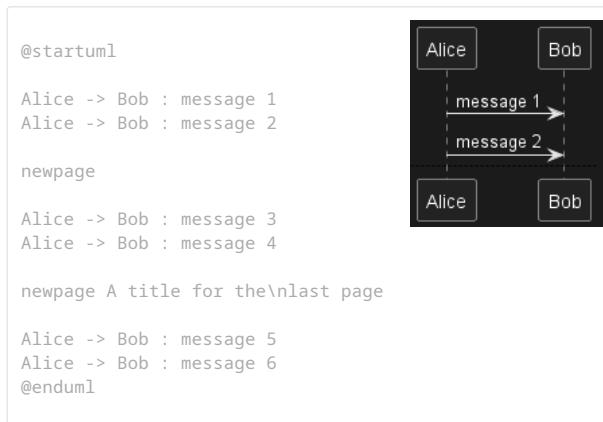
## Разбиение диаграм

Ключевое слово `newpage` используется для разбиения диаграм на несколько изображений.

Вы можете указать название страницы сразу после ключевого слова `newpage`. Это название заменяет указанное ранее название, если такое имеется.

Это очень полезно для печати длинных диаграмм на нескольких страницах.

(Примечание: это действительно работает. В превью показана только первая страница, но это из-за особенностей онлайн отображения.)



## Группировка сообщений

Группировать сообщения возможно используя следующие ключевые слова:

- `alt/else`
- `opt`
- `loop`
- `par`
- `break`
- `critical`
- `group`, соответствует тексту который должен быть отображен

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

группы').

Ключевое слово `end` используется для завершения группы.

Допустимо вложение группы в группу.

```
@startuml
Alice -> Bob: Authentication Request

alt successful case

    Bob -> Alice: Authentication Accepted

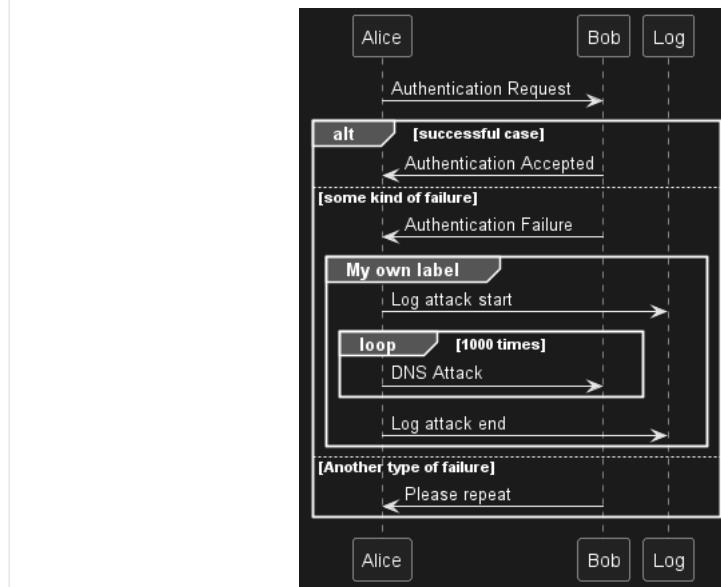
else some kind of failure

    Bob -> Alice: Authentication Failure
    group My own label
        Alice -> Log : Log attack start
        loop 1000 times
            Alice -> Bob: DNS Attack
        end
        Alice -> Log : Log attack end
    end

else Another type of failure

    Bob -> Alice: Please repeat

end
@enduml
```



## Дополнительная метка группы

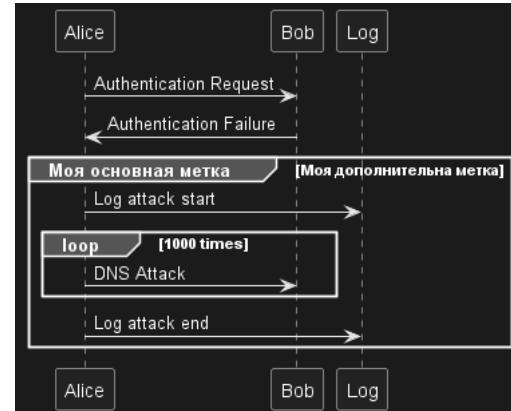
Для `group` можно добавить дополнительный текст или метку, которые будут отображаться в заголовке. Для этого их надо указать между квадратными скобками [ и ].

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```

Alice -> Bob: Authentication Request
Bob -> Alice: Authentication Failure
group Моя основная метка [Моя дополнительна метка]
    Alice -> Log : Log attack start
    loop 1000 times
        Alice -> Bob: DNS Attack
    end
    Alice -> Log : Log attack end
end
@enduml

```



[Ссылка QA-2503]

## Примечания в сообщениях

Можно помещать заметки к сообщениям, используя ключевые слова `note left` (заметка слева) или `note right` (заметка справа) сразу после сообщения.

Можно делать многострочные заметки используя ключевое слово `end note` для завершения заметки.

```

@startuml
Alice->Bob : привет
note left: это первая заметка

Bob->Alice : ага
note right: это другая заметка

Bob->Bob : я размышляю над этим
note left
заметки
могут занимать
несколько строчек
end note
@enduml

```



- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

Возможно размещение примечаний относительно участников с использованием ключевых слов note left of (разместить слева от), note right of (разместить права от) или note over (разместить над).

Возможно выделить примечание, изменив цвет фона.

Можно делать многострочные заметки используя ключевое слово end note для завершения заметки.

```
@startuml
participant Alice
participant Bob
note left of Alice #aqua
Будет отображено
слева от Алисы
end note

note right of Alice: Будет отображено справа от Алисы

note over Alice: Будет отображено над Алисой.

note over Alice, Bob #FFAAAA: Будет отображено\n над Алисой и Бобом.

note over Bob, Alice
А это еще
один пример
длинной заметки.
end note
@enduml
```



## Изменение формы примечаний

Вы можете использовать hnote и xnote для изменения формы примечаний:

- hnote для примечания в виде шестиугольника;
- xnote для примечания в виде прямоугольника.

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность    прецедентов    классов    активности    компонент    состояний

```

caller -> server : conReq
hnote over caller : idle
caller -> server : conConf
rnote over server
  "r" as rectangle
  "h" as hexagon
endrnote
rnote over server
  что-то важное
  на нескольких
  строках
endrnote
hnote over caller
  что-то не важное
  на нескольких
  строках
endhnote
@enduml

```



[Ref. QA-1765]

## Заметка отображаемая поверх всех участников

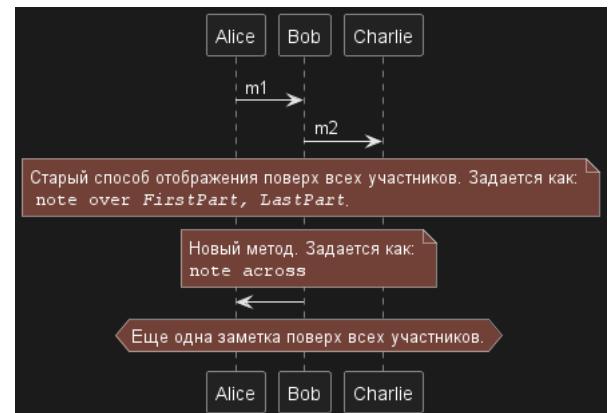
Вы можете сделать заметку, отображаемую поверх всем участникам, используя синтаксис:

- note across: note\_description

```

@startuml
Alice->Bob:m1
Bob->Charlie:m2
note over Alice, Charlie: Старый способ отображения поверх всех участников
note across: Новый метод. Задается как:\n"note across"
Bob->Alice
hnote across: Еще одна заметка поверх всех участников.
@enduml

```



[Ref. QA-9738]

## Выравнивание нескольких заметок по одному уровню [/]

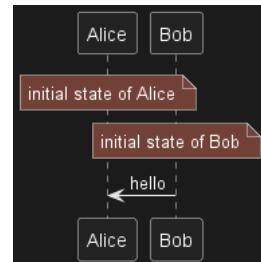
Вы можете выровнить несколько заметок по одному уровню используя синтаксис / :

- без / (по умолчанию заметки не выравниваются)

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

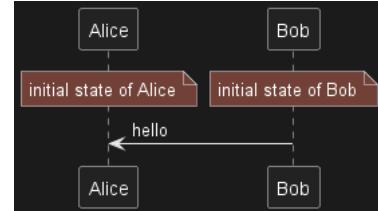
последовательность    прецедентов    классов    активности    компонент    состояний

```
note over Alice : initial state of Alice
note over Bob : initial state of Bob
Bob -> Alice : hello
@enduml
```



- C / (заметки выровнены)

```
@startuml
note over Alice : initial state of Alice
/ note over Bob : initial state of Bob
Bob -> Alice : hello
@enduml
```



[Ref. [QA-354](#)]

## Creole и HTML

Так же можно использовать форматирование на Creole:

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность прецедентов классов активности компонент состояний

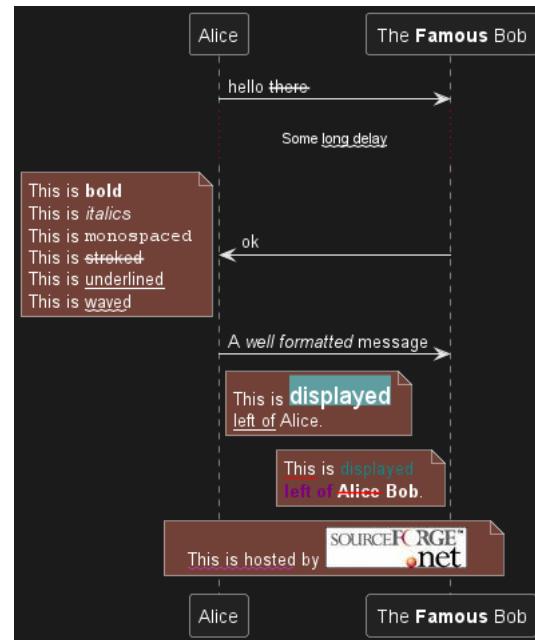
```

participant Alice
participant "The **Famous** Bob" as Bob

Alice -> Bob : hello --there--
... Some ~~long delay~~ ...
Bob -> Alice : ok
note left
    This is **bold**
    This is //italics//
    This is ""monospaced"""
    This is --stroked--
    This is __underlined__
    This is ~~waved~~
end note

Alice -> Bob : A //well formatted// message
note right of Alice
    This is <back:cadetblue><size:18>displayed</size></back>
    __left of__ Alice.
end note
note left of Bob
    <u:red>This</u> is <color #118888>displayed</color>
    **<color purple>left of</color> <s:red>Alice</strike> Bob**.
end note
note over Alice, Bob
    <w:#FF33FF>This is hosted</w> by <img sourceforge.jpg>
end note
@enduml

```



## Разделитель

Вы можете использовать разделитель == , чтобы разбить диаграмму на несколько логических этапов.

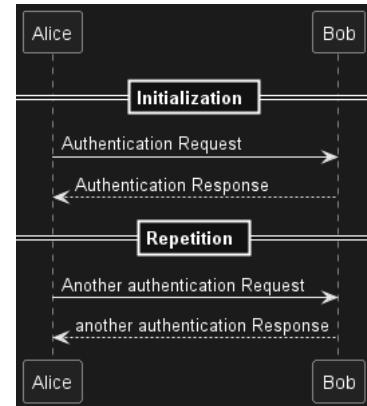
- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [FAQ](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```
== Initialization ==
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

== Repetition ==
Alice -> Bob: Another authentication Request
Alice <-- Bob: another authentication Response

@enduml
```



## Ссылки

Вы можете использовать ссылки в диаграммах с помощью ключевого слова `ref over`.

```
@startuml
participant Alice
actor Bob

ref over Alice, Bob : инициализация

Alice -> Bob : hello

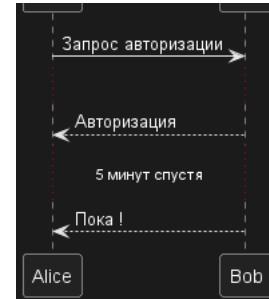
ref over Bob
    Тут мы можем занять
    несколько строк
end ref
@enduml
```



## Задержка на диаграммах

Вы можете использовать конструкцию `... для представления временной задержки в процессе на диаграмме. При необходимости можно снабдить задержку комментарием.`

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

[последовательность](#)
[прецедентов](#)
[классов](#)
[активности](#)
[компонент](#)
[состояний](#)


## Перенос текста

По умолчанию текст сообщения отображается в одну строку.

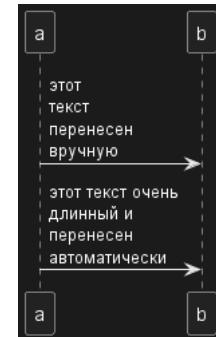
Перенос текста для его отображения на нескольких строках можно сделать:

- вручную, добавив \n в месте разрыва строки;
- автоматически, установив максимальное количество символов в строке с использованием параметра maxMessageSize . При этом перенос осуществляется по словам.

```

@startuml
skinparam maxMessageSize 100
participant a
participant b
a -> b :этот\птекст\пперенесен\повручную
a -> b :этот текст очень длинный и перенесен автоматически
@enduml

```



## Промежутки

Вы можете использовать ||| чтобы показать промежутки в диаграммах..

Так же возможно указать промежуток в пикселях.



## Активация и деактивация линии существования

`activate` и `deactivate` используются чтобы обозначить активацию участника.

Линия существования появляется в момент активации участника.

`activate` и `deactivate` применяются к предыдущему сообщению.

`destroy` обозначает конец линии существования участника.

```

@startuml
participant User

User -> A: DoWork
activate A

A -> B: << createRequest >>
activate B

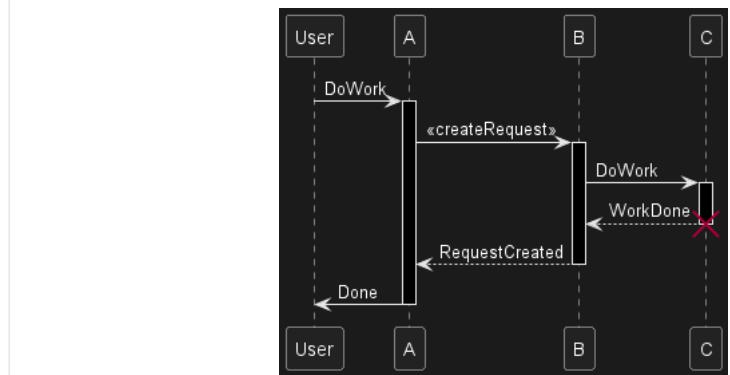
B -> C: DoWork
activate C
C --> B: WorkDone
destroy C

B --> A: RequestCreated
deactivate B

A -> User: Done
deactivate A

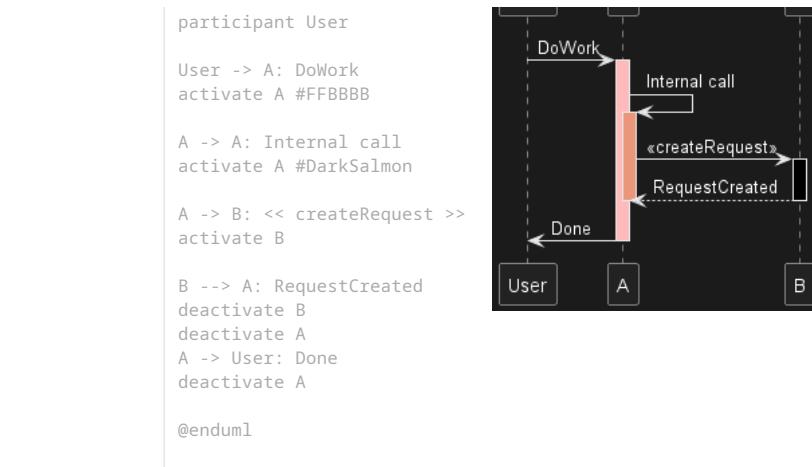
@enduml

```



Можно использовать вложенные линии существования, и можно указывать цвет для этих линий.

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

[последовательность](#)
[прецедентов](#)
[классов](#)
[активности](#)
[компонент](#)
[состояний](#)


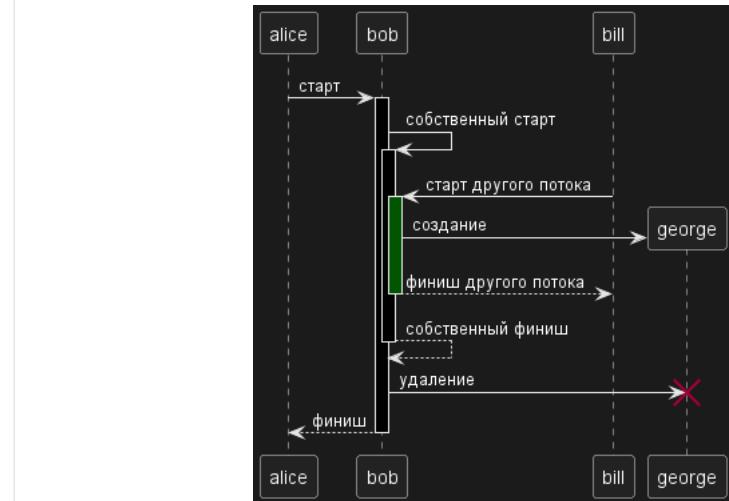
Autoactivation is possible and works with the return keywords: Автоактивация также возможна, для этого используйте команду autoactivate on . Для определения конца активации используются команды return :

```

@startuml
autoactivate on
alice -> bob : старт
bob -> bob : собственный старт
bill -> bob #005500 : старт другого потока
bob -> george ** : создание
return финиш другого потока
return собственный финиш
bob -> george !! : удаление
return финиш

@enduml

```



## Return

Команда return генерирует сообщение возврата с необязательной текстовой меткой.

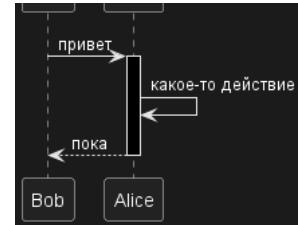
Точки возврата являются та линия, которая вызвала последнюю активацию текущей линии жизни.

Синтаксис - return label , где label , если указано, является любой строкой, приемлемой для обычных сообщений.

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```
Bob -> Alice : привет
activate Alice
Alice -> Alice : какое-то действие
return пока
@enduml
```



## Отображение создания участника процессом

Вы можете использовать ключевое слово `create` перед декларацией сообщения для акцентирования факта, что принимающий участник *создается* данным сообщением.

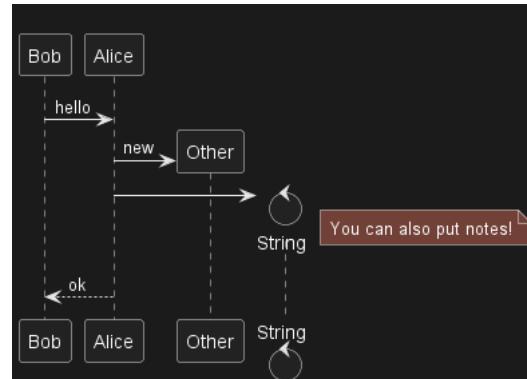
```
@startuml
Bob -> Alice : hello

create Other
Alice -> Other : new

create control String
Alice -> String
note right : You can also put notes!

Alice --> Bob : ok

@enduml
```



## Быстрый синтаксис для активации, деактивации и создания

Сразу после указания целевого участника можно использовать следующий синтаксис:

- `++` Активировать цель (опционально за этим может следовать [цвет](#))
- `--` Деактивировать источник
- `**` Создать экземпляр цели
- `!!` Уничтожить экземпляр цели

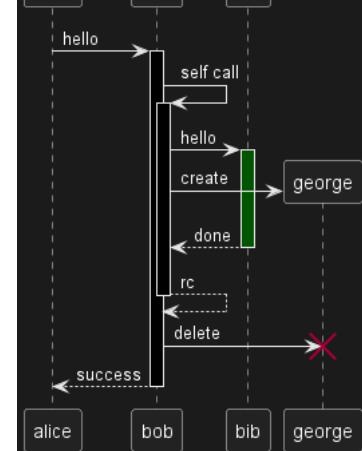
- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```

alice -> bob ++ : hello
bob -> bob ++ : self call
bob -> bib ++ #005500 : hello
bob -> george ** : create
return done
return rc
bob -> george !! : delete
return success
@enduml

```

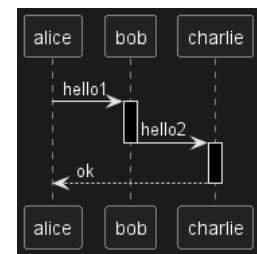


Затем вы можете смешивать активацию и деактивацию в одной строке

```

@startuml
alice -> bob ++ : hello1
bob -> charlie ---+ : hello2
charlie --> alice -- : ok
@enduml

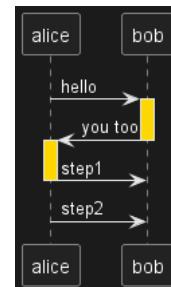
```



```

@startuml
@startuml
alice -> bob ---+ #gold: hello
bob -> alice ---+ #gold: you too
alice -> bob --: step1
alice -> bob : step2
@enduml
@enduml

```

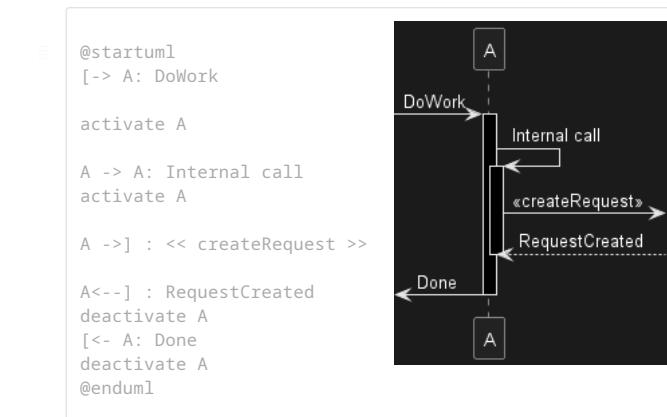


[Ссылка на [QA-4834](#), [QA-9573](#) и [QA-13234](#)]

## Входящие и исходящие сообщения

Вы можете использовать входящие или исходящие стрелки если вы хотите сфокусироваться на части диаграммы.

Используйте квадратные скобки для указания левой " [ " или правой " ] " стороны диаграммы



Вы также можете использовать следующий синтаксис:

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

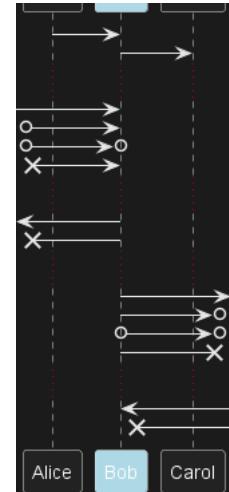


```

participant Alice
participant Bob #lightblue
Alice -> Bob
Bob -> Carol
...
[ -> Bob
[ o-> Bob
[ o->o Bob
[ x-> Bob
...
[ <- Bob
[ x<- Bob
...
Bob ->]
Bob ->o]
Bob o->o]
Bob ->x]
...
Bob <-]
Bob x<-]

@enduml

```



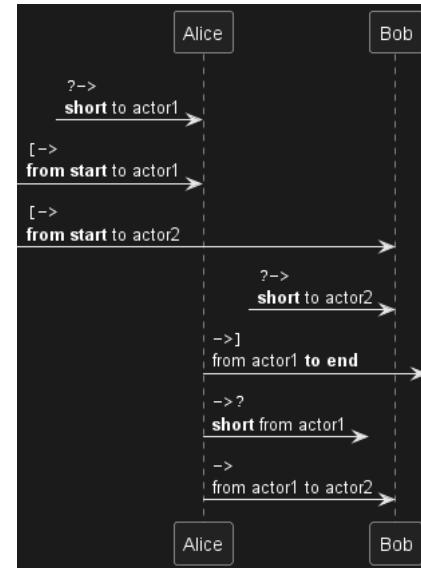
## Короткие стрелки для входящих и исходящих сообщений

Вы можете использовать **короткие** стрелки с помощью ?

```

@startuml
?-> Alice : ""?->""\n**short** to actor1
[-> Alice : ""[->""\n**from start** to actor1
[-> Bob : ""[->""\n**from start** to actor2
?-> Bob : ""?->""\n**short** to actor2
Alice ->] : ""->]""\nfrom actor1 **to end**
Alice ->? : ""->?""\n**short** from actor1
Alice -> Bob : ""->""\nfrom actor1 to actor2
@enduml

```



[Ссылка на QA-310]

## Якоря и длительность

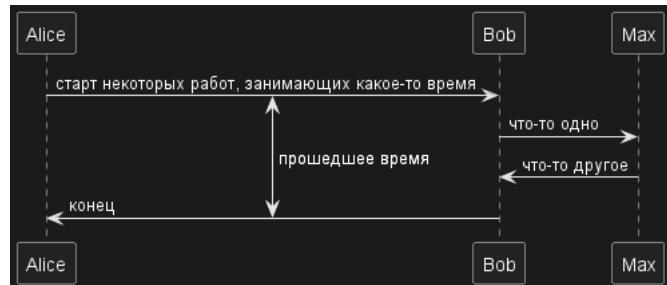
- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

```
@startuml
!pragma teoz true

{start} Alice -> Bob : старт некоторых работ, занимающих какое-то время
Bob -> Max : что-то одно
Max -> Bob : что-то другое
{end} Bob -> Alice : конец

{start} <-> {end} : прошедшее время

@enduml
```



Для указания прагмы можно использовать опцию [командной строки](#) `-Pteoz=true`

```
java -jar plantuml.jar -Pteoz=true
```

[Ref. [issue-582](#)]

## Шаблоны и отметки

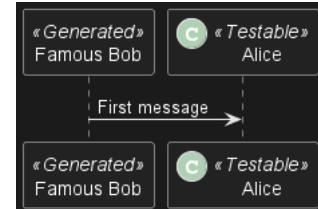
Можно добавить шаблоны к участникам используя `<<` и `>>`.

В шаблоне вы можете добавить отмеченного участника в цветном круге используя синтаксис `(X,color)`.

```
@startuml
participant "Famous Bob" as Bob << Generated >>
participant Alice << (C,#ADD1B2) Testable >>

Bob->>Alice: First message

@enduml
```



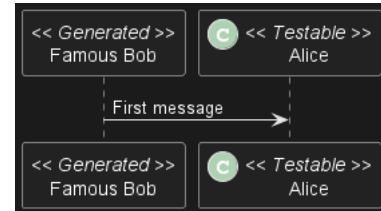
По умолчанию, символ *guillemet* используется для отображения шаблона. Вы можете изменить это поведение, используя `skinparam guillemet`:

- В начало
- Что нового?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam guillemet false
participant "Famous Bob" as Bob << Generated >>
participant Alice << (C,#ADD1B2) Testable >>

Bob->Alice: First message

@enduml
```

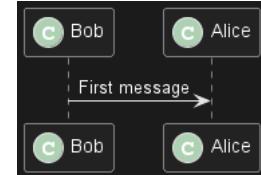


```
@startuml

participant Bob << (C,#ADD1B2) >>
participant Alice << (C,#ADD1B2) >>

Bob->Alice: First message

@enduml
```



## Больше информации в заголовках

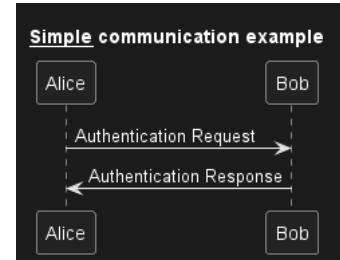
Вы можете использовать форматирование на [Creole](#) для заголовков.

```
@startuml

title __Simple__ **communication** example

Alice -> Bob: Authentication Request
Bob -> Alice: Authentication Response

@enduml
```



С помощью последовательности символов \n вы можете добавить перевод строки в заголовок.

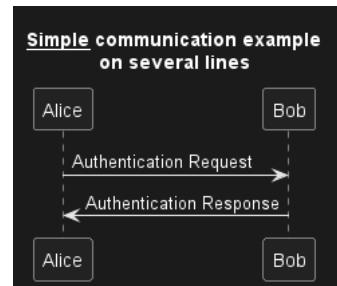
- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

```
title __Simple__ communication example\nnon several lines
```

```
Alice -> Bob: Authentication Request  
Bob -> Alice: Authentication Response
```

```
@enduml
```



Вы также можете задать заголовок на нескольких строках, используя ключевые слова title И end title .

```
@startuml

title
<u>Simple</u> communication example
on <i>several</i> lines and using <font color=red>html</font>
This is hosted by <img:sourceforge.jpg>
end title

Alice -> Bob: Authentication Request
Bob -> Alice: Authentication Response

@enduml
```



## Группировка участников

Можно создать прямоугольник вокруг участников, используя команды box И end box .

Вы можете задать опциональный заголовок и цвет фона, после команды the box .

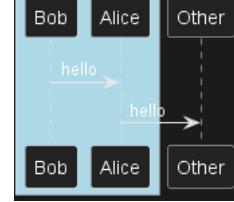
-  В начало
-  Что нового ?
-  Быстрый старт
-  Online Server
-  Запуск
-  F.A.Q.
-  Скачать
-  Форум
-  Theme
-  Препроцессинг
-  Стандартная библиотека
-  Hitchhiker's Guide
-  PDF Guide

последовательность    прецедентов    классов    активности    компонент    состояний

```
box "Internal Service" #LightBlue
participant Bob
participant Alice
end box
participant Other

Bob -> Alice : hello
Alice -> Other : hello

@enduml
```



WARNING This translation need to be updated. WARNING

## Удаление футера

Вы можете использовать ключевое слово `hide footbox` для удаления футера из диаграммы.

```
@startuml
hide footbox
title Footer removed

Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

@enduml
```



## Skinparam

Вы можете использовать команду `skinparam` для изменения шрифтов и цветов диаграммы

Вы можете использовать данную команду :

- В определении диаграммы, как любую другую команду,
- В **подключенном файле**,
- В конфигурационном файле, указанном в **командной строке** в **задании ANT**.

Вы можете изменить другие параметры отображения, как видно из следующих примеров:

- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

```

skinparam sequenceArrowThickness 2
skinparam roundcorner 20
skinparam maxmessagessize 60
skinparam sequenceParticipant underline

actor User
participant "First Class" as A
participant "Second Class" as B
participant "Last Class" as C

User -> A: DoWork
activate A

A -> B: Create Request
activate B

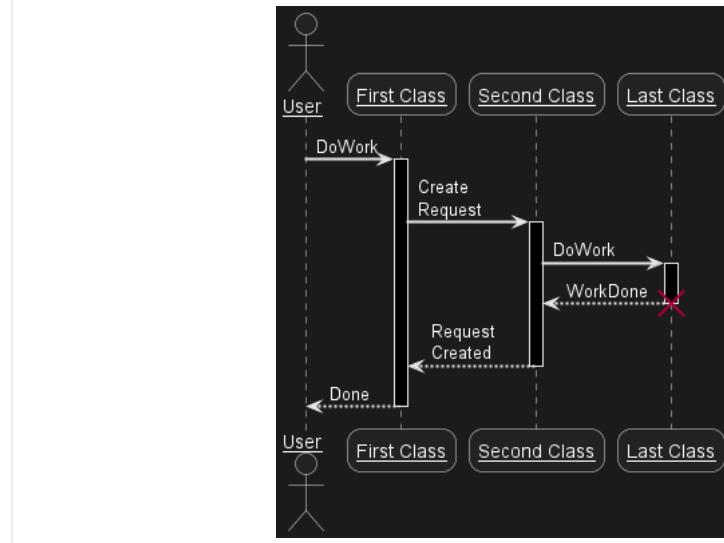
B -> C: DoWork
activate C
C --> B: WorkDone
destroy C

B --> A: Request Created
deactivate B

A --> User: Done
deactivate A

@enduml

```



- [В начало](#)
- [Что нового?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

```

skinparam backgroundColor #EEEBCD
skinparam handwritten true

skinparam sequence {
    ArrowColor DeepSkyBlue
    ActorBorderColor DeepSkyBlue
    LifeLineBorderColor blue
    LifeLineBackgroundColor #A9DCDF
}

ParticipantBorderColor DeepSkyBlue
ParticipantBackgroundColor DodgerBlue
ParticipantFontName Impact
ParticipantFontSize 17
ParticipantFontColor #A9DCDF

ActorBackgroundColor aqua
ActorFontColor DeepSkyBlue
ActorFontSize 17
ActorFontName Aapex
}

actor User
participant "First Class" as A
participant "Second Class" as B
participant "Last Class" as C

User -> A: DoWork
activate A

A -> B: Create Request
activate B

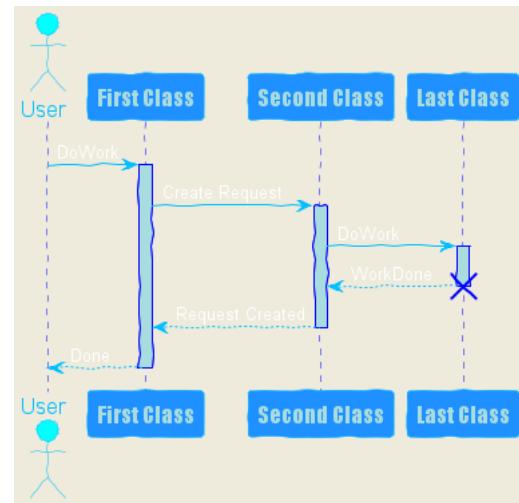
B -> C: DoWork
activate C
C --> B: WorkDone
destroy C

B --> A: Request Created
deactivate B

A --> User: Done
deactivate A

@enduml

```



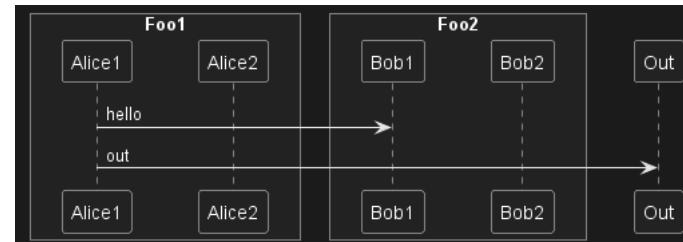
## Изменение отступов

Вы можете изменить некоторые настройки отступов

- В начало
- Что нового?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam ParticipantPadding 20
skinparam BoxPadding 10

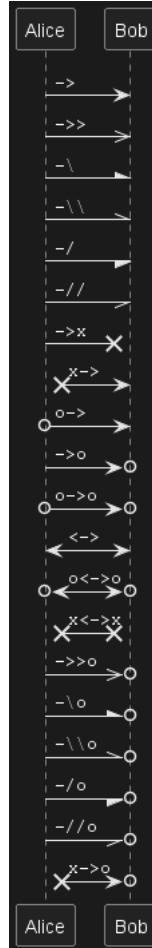
box "Foo1"
participant Alice1
participant Alice2
end box
box "Foo2"
participant Bob1
participant Bob2
end box
Alice1 -> Bob1 : hello
Alice1 -> Out : out
@enduml
```



## Приложение: Примеры всех типов стрелок

### Обычная стрелка

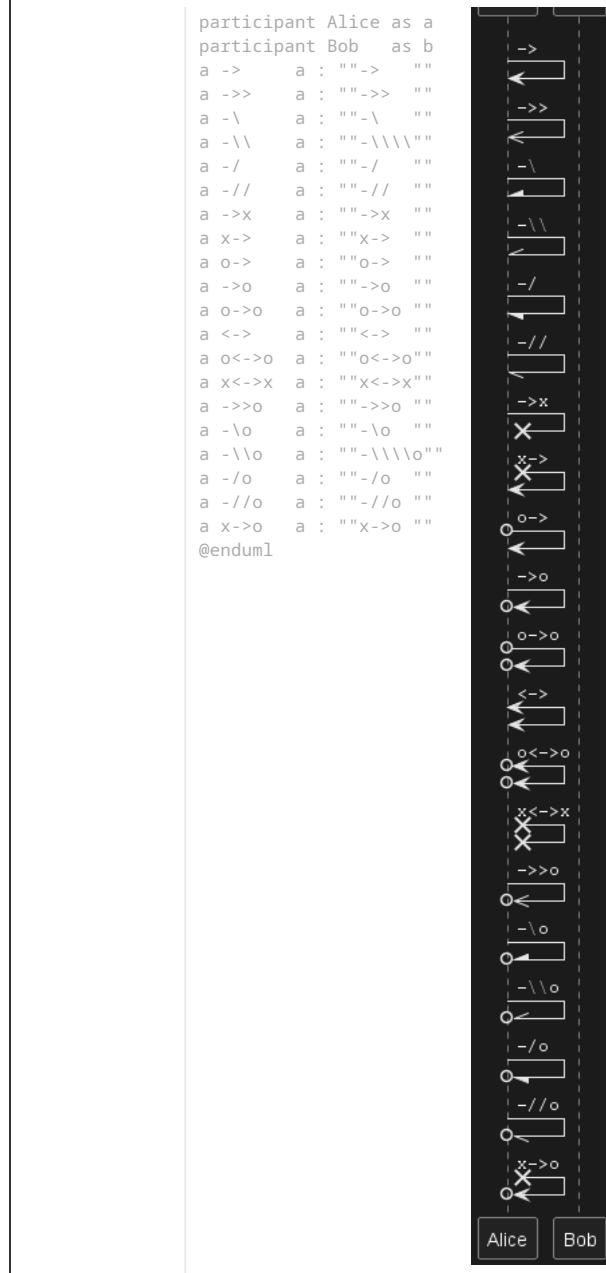
```
@startuml
participant Alice as a
participant Bob as b
a -> b : ""-> ""
a ->> b : ""->> ""
a -\ b : ""-\ ""
a -\\" b : ""-\\"\\"
a -/ b : ""-/ ""
a -// b : ""-// ""
a ->x b : ""->x ""
a x-> b : ""x-> ""
a o-> b : ""o-> ""
a ->o b : ""->o ""
a o->o b : ""o->o ""
a <-> b : ""<-> ""
a o<->o b : ""o<->o ""
a x<->x b : ""x<->x""
a ->>o b : ""->>o ""
a -\o b : ""-\o ""
a -\\"o b : ""-\\"\\o"
a -/o b : ""-/o ""
a -//o b : ""-//o ""
a x->o b : ""x->o ""
@enduml
```



### Возвращающаяся стрелка

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний

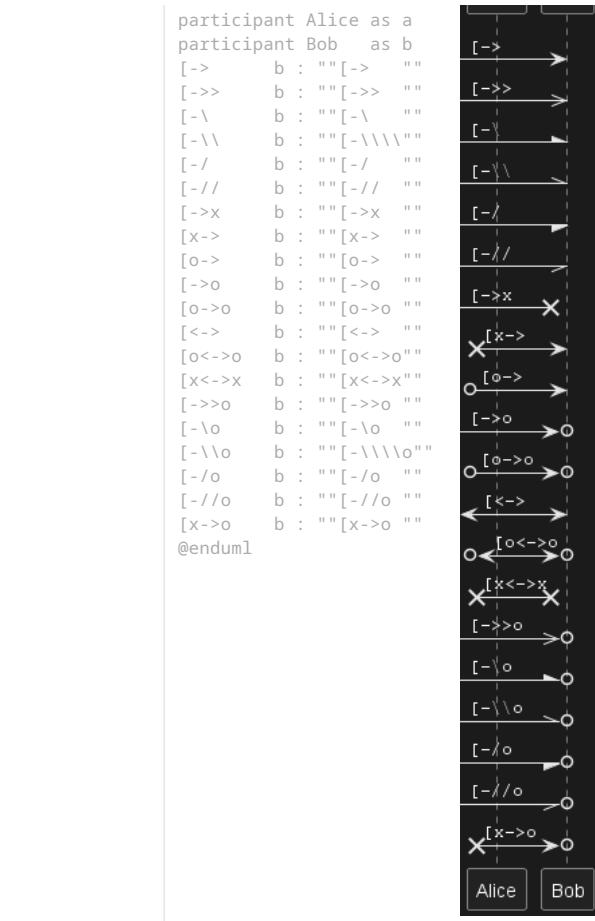


Входящие и исходящие сообщения (с '[' , ']')

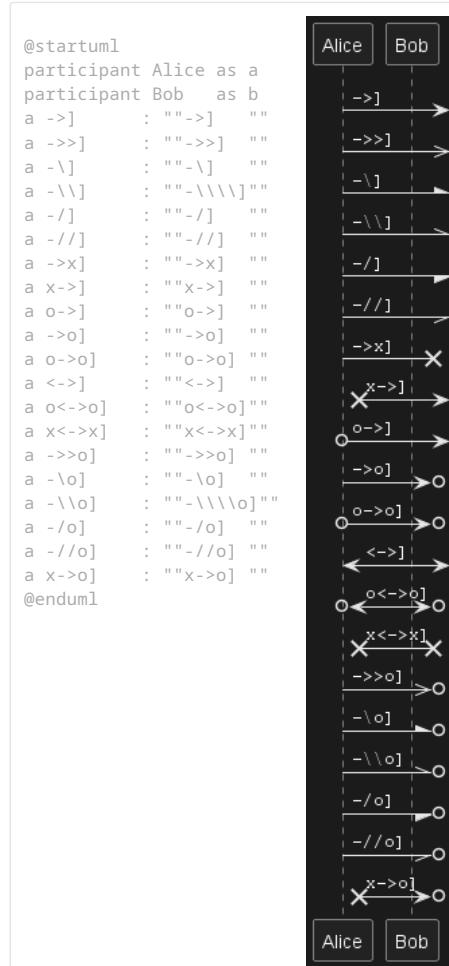
Входящие сообщения (с '[')

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

последовательность    прецедентов    классов    активности    компонент    состояний



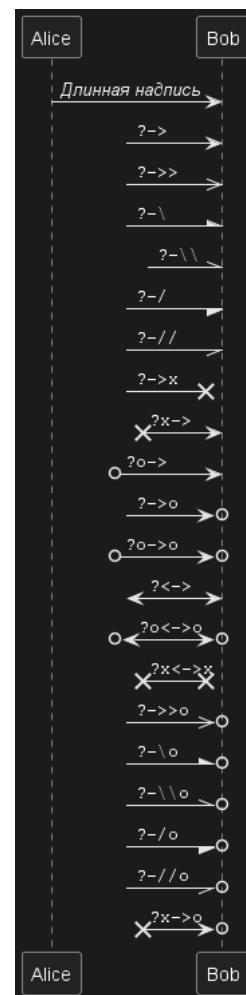
### Исходящие сообщения (с '])'



- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

### Короткие входящие (с '?')

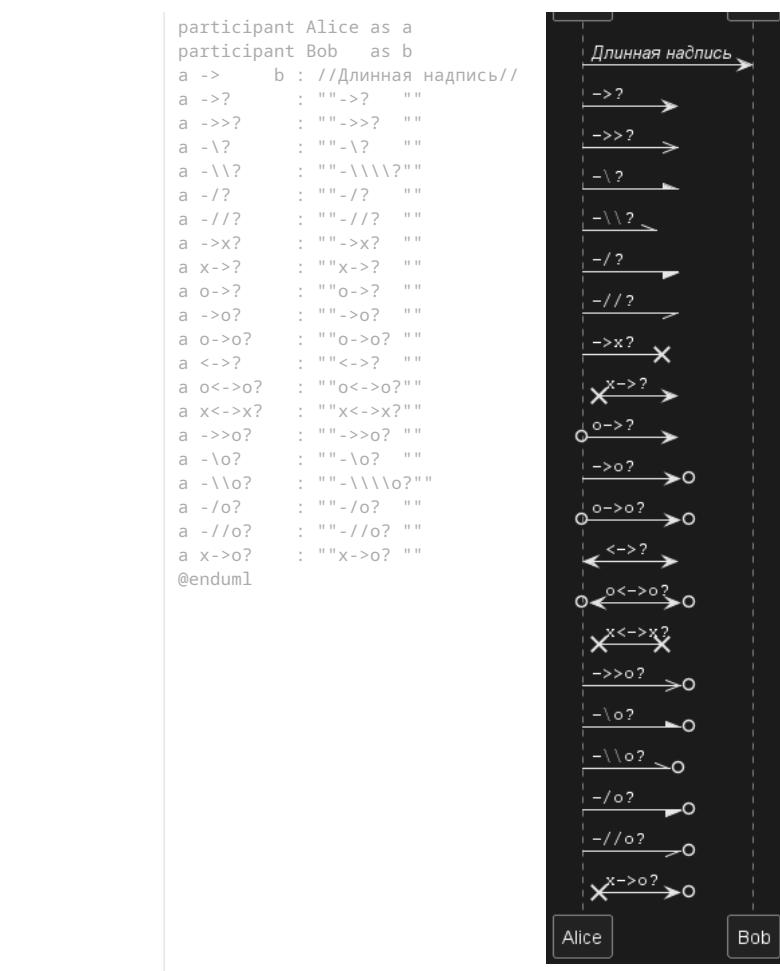
```
@startuml
participant Alice as a
participant Bob as b
a -> b : //Длинная надпись//
?-> b : "?-> "
?->> b : "?->> "
?-\\ b : "?-<\\" "
?-\\ b : "?->\\\" "
?-/ b : "?->/ "
?-// b : "?->// "
?->x b : "?->x "
?x-> b : "?x-> "
?o-> b : "?o-> "
?-o b : "?->o "
?o->o b : "?o->o "
?<-> b : "?<-> "
?o<->o b : "?o<->o "
?x<->x b : "?x<->x "
?->o b : "?->o "
?-\\o b : "?->\\o "
?-\\o b : "?->\\o "
?-//o b : "?->//o "
?-x>o b : "?->x>o "
@enduml
```



### Короткие исходящие (с '?')

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность    прецедентов    классов    активности    компонент    состояний



## Разные стили отображения

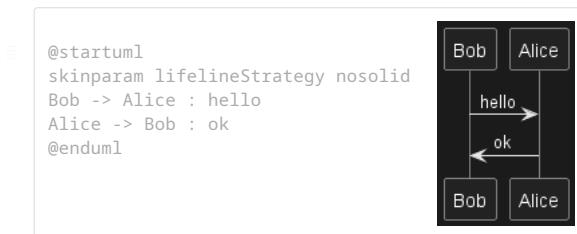
### По умолчанию



### LifelineStrategy

- nosolid

Этот стиль и используется по умолчанию



[Ссылка на QA-9016]

- solid

Для того, чтобы сделать вертикальные линии сплошными используйте команду  
skinparam lifelineStrategy solid

-  В начало
-  Что нового ?
-  Быстрый старт
-  Online Server
-  Запуск
-  F.A.Q.
-  Скачать
-  Форум
-  Theme
-  Препроцессинг
-  Стандартная библиотека
-  Hitchhiker's Guide
-  PDF Guide

последовательность    прецедентов    классов    активности    компонент    состояний

```
skinparam lifelineStrategy solid
Bob -> Alice : hello
Alice -> Bob : ok
@enduml
```



[Ссылка на QA-2794]

### style strictuml

Для отображения в строгом соответствии с UML (также меняется стиль стрелок: кончики отображаются как треугольники), вы можете использовать

- skinparam style strictuml

```
@startuml
skinparam style strictuml
Bob -> Alice : hello
Alice -> Bob : ok
@enduml
```



[ссылка QA-1047]

## Скрытие ни с кем не связанных участников

По умолчанию отображаются все участники.

```
@startuml
participant Alice
participant Bob
participant Carol
Alice -> Bob : Привет
@enduml
```



Но Вы можете скрыть таких участников используя команду `hide unlinked`.

```
@startuml
hide unlinked
participant Alice
participant Bob
participant Carol
Alice -> Bob : Привет
@enduml
```



[Ref. QA-4247]

## Цвет сгруппированных сообщений

Можно задать `цвет` для сгруппированных сообщений:

- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

```

Alice -> Bob: Запрос авторизации
alt#Gold #LightBlue Успешный кейс
    Bob -> Alice: Запрос удовлетворен
else #Pink Ошибка
    Bob -> Alice: Запрос отклонен
end
@enduml

```



[Ref. [QA-4750](#) and [QA-6410](#)]

## Основной фрейм

```

@startuml
mainframe Это **основной фрейм**
Alice->Bob : Привет
@enduml

```



[Ref. [QA-4019](#) and [Issue#148](#)]

## Slanted or odd arrows

You can use the `(nn)` option (before or after arrow) to make the arrows slanted, where `nn` is the number of shift pixels.

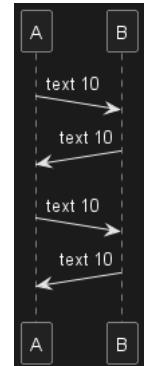
[Available only after v1.2022.6beta+]

```

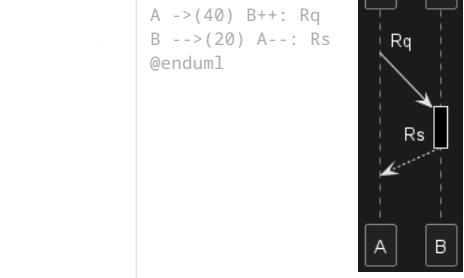
@startuml
A ->(10) B: text 10
B ->(10) A: text 10

A ->(10) B: text 10
A (10)<- B: text 10
@enduml

```

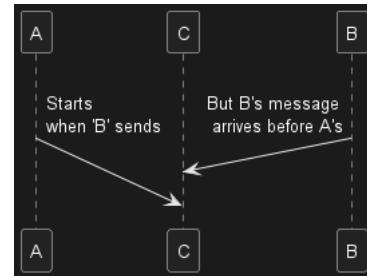


- [В начало](#)
- [Что нового ?](#)
- [Быстрый старт](#)
- [Online Server](#)
- [Запуск](#)
- [F.A.Q.](#)
- [Скачать](#)
- [Форум](#)
- [Theme](#)
- [Препроцессинг](#)
- [Стандартная библиотека](#)
- [Hitchhiker's Guide](#)
- [PDF Guide](#)

[последовательность](#) [прецедентов](#) [классов](#) [активности](#) [компонент](#) [состояний](#)


[Ref. QA-14145]

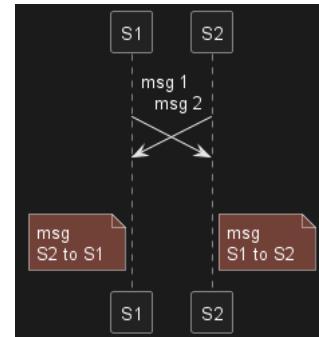
```
@startuml
!pragma teoz true
A ->(50) C: Starts\nwhen 'B' sends
& B ->(25) C: \nBut B's message\n arrives before A's
@enduml
```



[Ref. QA-6684]

```
@startuml
!pragma teoz true
S1 ->(30) S2: msg 1\n
& S2 ->(30) S1: msg 2

note left S1: msg\nS2 to S1
& note right S2: msg\nS1 to S2
@enduml
```



[Ref. QA-1072]

[Privacy Policy](#) [Advertise](#)