
Рисование UML с PlantUML
Справочник языка
(Версия 5737)

PlantUML – проект с открытым исходным кодом, который позволяет быстро рисовать диаграммы:

- Диаграмма последовательности,
- Диаграмма прецедентов,
- Диаграмма классов,
- Диаграмма деятельности,
- Диаграмма компонентов,
- Диаграмма состояний,
- Диаграмма объектов.

Диаграммы создаются, используя простой и интуитивный язык.

Оглавление

1	Диаграмма последовательности	4
1.1	Основные понятия	4
1.2	Объявление объектов	5
1.3	Спецсимволы в названии объектов	6
1.4	Сообщения, посылаемые самому себе	6
1.5	Нумерация последовательности сообщений	7
1.6	Заголовок	9
1.7	Разделение схем	9
1.8	Группировка сообщений	10
1.9	Примечания к сообщениям	12
1.10	Другие примечания	13
1.11	Форматирование с помощью HTML	14
1.12	Разделитель	15
1.13	Линии жизни. Активация и завершение	16
1.14	Создание участников	18
1.15	Входящие и исходящие сообщения	19
1.16	Стереотипы и иконки	20
1.17	Больше информации о заголовках	21
1.18	Выделение участников	23
1.19	Удаление нижнего колонтитула	24
1.20	Команда «skinparam»	25
1.21	Команда «skin»	27
2	Диаграмма прецедентов	28
2.1	Прецеденты	28
2.2	Экторы	29
2.3	Примеры	30
2.4	Наследование или обобщение	31
2.5	Примечания	32
2.6	Стереотипы	33
2.7	Изменение направления стрелок	34
2.8	Название диаграммы	35
2.9	Направление диаграммы	36
2.10	Команда «skinparam»	37

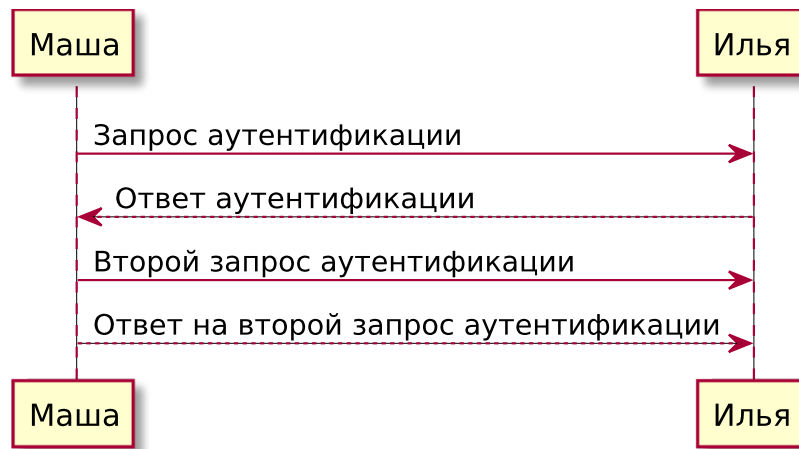
3	Диаграмма классов	38
3.1	Связи между классами	38
3.2	Пометки на связях	39
3.3	Добавление методов	40
3.4	Определение видимости	41
3.5	Заметки и стереотипы	42
3.6	Дополнительно о заметках	43
3.7	Абстрактные классы и интерфейсы	44
3.8	Использование спецсимволов	45
3.9	Скрытые атрибуты и методы	46
3.10	Специальные значки	47
3.11	Пакеты	48
3.12	Пространство имен	49
3.13	Изменение направления стрелок	50
3.14	Интерфейс «lollipop»	51
3.15	Заголовок диаграммы	51
3.16	Ассоциация классов	52
3.17	Команда «skinparam»	53
3.18	Разделение больших диаграмм	54
4	Диаграмма деятельности	55
4.1	Простая деятельность	55
4.2	Комментарии к стрелкам	55
4.3	Изменение направления стрелок	56
4.4	Разветвление	57
4.5	Дополнительно про разветвление	58
4.6	Синхронизация	59
4.7	Полное описание деятельности	60
4.8	Заметки	61
4.9	Разделение	62
4.10	Название диаграммы	63
4.11	Команда «skinparam»	64
4.12	Итоговый пример	65

1. Диаграмма последовательности

1.1 Основные понятия

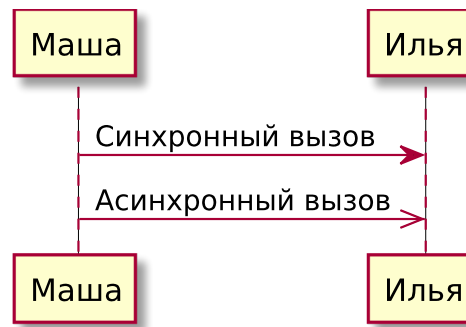
Описание каждой UML диаграммы должно начинаться с команды `@startuml` и заканчиваться `@enduml`. Последовательность «-»> используется для написания сообщения между объектами. Объекты не должны быть явно объявлены. Чтобы нарисовать прерывистую стрелку используйте «-->». Также возможно использование таких обозначений как «<-» и «<--». Это никак не повлияет на смысл диаграммы, но может улучшить читаемость. Пример:

```
@startuml
Маша -> Илья: Запрос аутентификации
Илья --> Маша: Ответ аутентификации
Маша -> Илья: Второй запрос аутентификации
Илья <-- Маша: Ответ на второй запрос аутентификации
@enduml
```



Чтобы использовать асинхронные сообщения, вы можете использовать «->>» или «<<-».

```
@startuml
Маша ->> Илья: Синхронный вызов
Маша ->> Илья: Асинхронный вызов
@enduml
```

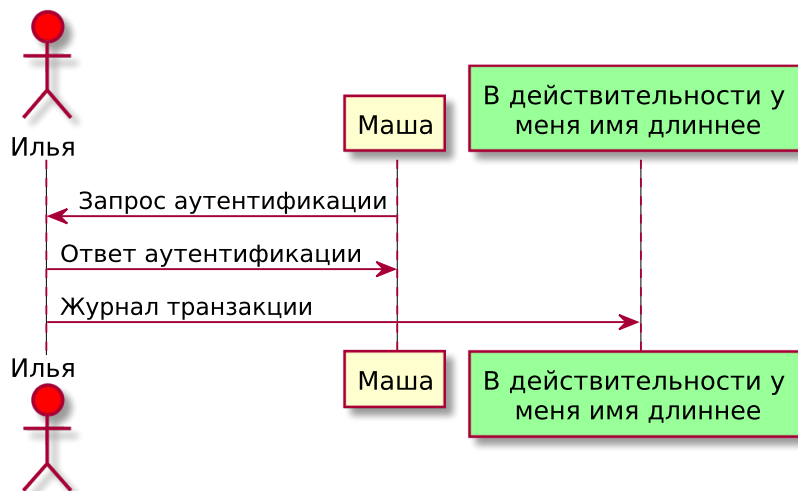


1.2 Объявление объектов

Можно изменить порядок участников с помощью ключевого слова «**participant**». Кроме того, для обозначения объекта можно использовать команду «**actor**», чтобы нарисовать человечка вместо окошка. Вы можете переименовать участника, используя ключевое слово «**as**». Также можно изменить цвет фона актера или объекта, используя HTML код или просто название цвета. Все, что начинается с кавычки ' является комментарием.

```

@startuml
actor Илья #red
' Разница между актером и участником изображена на диаграмме
participant Маша
participant "В действительности у \n меня имя длиннее" as L #99FF99
Маша->>Илья: Запрос аутентификации
Илья->>Маша: Ответ аутентификации
Илья->>L: Журнал транзакции
@enduml
  
```



1.3 Использование специальных символов в названии объектов

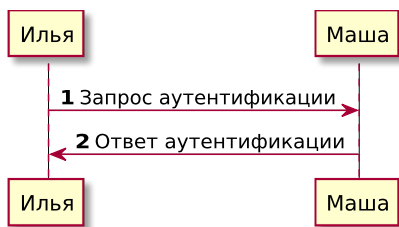
Вы можете использовать кавычки в названии объектов. Так же возможно использование ключевых слов в качестве имени объекта.

```
@startuml
Маша -> "Илья()" : Привет
"Илья()" -> "Оно очень \n длинное" as Длинное
' Вы так же можете объявить:
' "Илья()" -> Длинное as "Оно очень \n длинное"
Длинное --> "Илья()" : Хорошо
@enduml
```

1.5 Нумерация последовательности сообщений

Ключевое слово «autonumber» используется для нумерации сообщений.

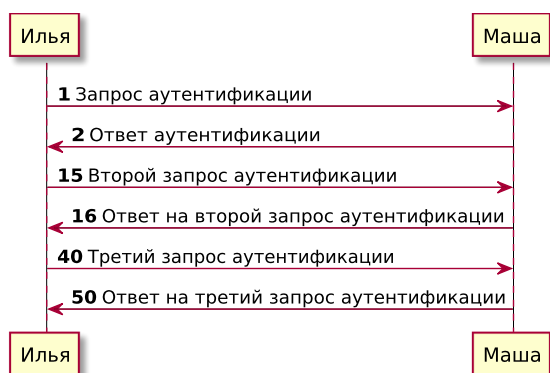
```
@startuml
autonumber
Илья -> Маша : Запрос аутентификации
Илья <- Маша : Ответ аутентификации
@enduml
```



Вы можете указать

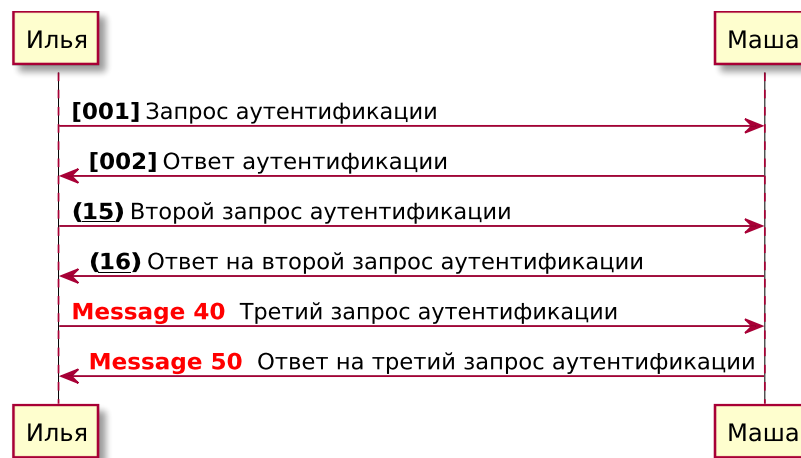
- начало нумерации «autonumber 'start'»,
- инкремент «autonumber 'start' 'increment'»

```
@startuml
autonumber
Илья -> Маша : Запрос аутентификации
Илья <- Маша : Ответ аутентификации
autonumber 15
Илья -> Маша : Второй запрос аутентификации
Илья <- Маша : Ответ на второй запрос аутентификации
autonumber 40 10
Илья -> Маша : Третий запрос аутентификации
Илья <- Маша : Ответ на третий запрос аутентификации
@enduml
```



Вы можете указать формат числа при помощи двойных кавычек. Применяется форматирование подобное классу Java `DecimalFormat` («0» означает цифру, «#» означает как цифру так и ноль, если не определено.) Вы можете также использовать некоторые html-тэги в формате.

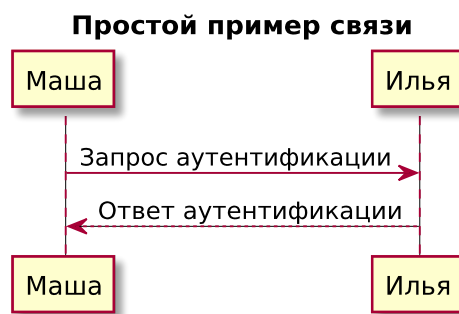
```
@startuml
autonumber "<b>[000]"
Илья -> Маша : Запрос аутентификации
Илья <- Маша : Ответ аутентификации
autonumber 15 "<b>(<u>##</u>)"
Илья -> Маша : Второй запрос аутентификации
Илья <- Маша : Ответ на второй запрос аутентификации
autonumber 40 10 "<font color=red><b>Message 0"
Илья -> Маша : Третий запрос аутентификации
Илья <- Маша : Ответ на третий запрос аутентификации
@enduml
```



1.6 Заголовок

Ключевое слово «`title`» используется, чтобы обозначить заголовок.

```
@startuml
title Простой пример связи
Маша -> Илья: Запрос аутентификации
Илья --> Маша: Ответ аутентификации
@enduml
```



1.7 Разделение схем

Ключевое слово «`newpage`» используется, чтобы разделить схему на несколько изображений. Вы можете поместить заголовок для новой страницы сразу после ключевого слова «`newpage`». Это применяется для распечатки длинных схем на нескольких страницах.

```
@startuml
Маша -> Илья : Сообщение 1
Маша -> Илья : Сообщение 2

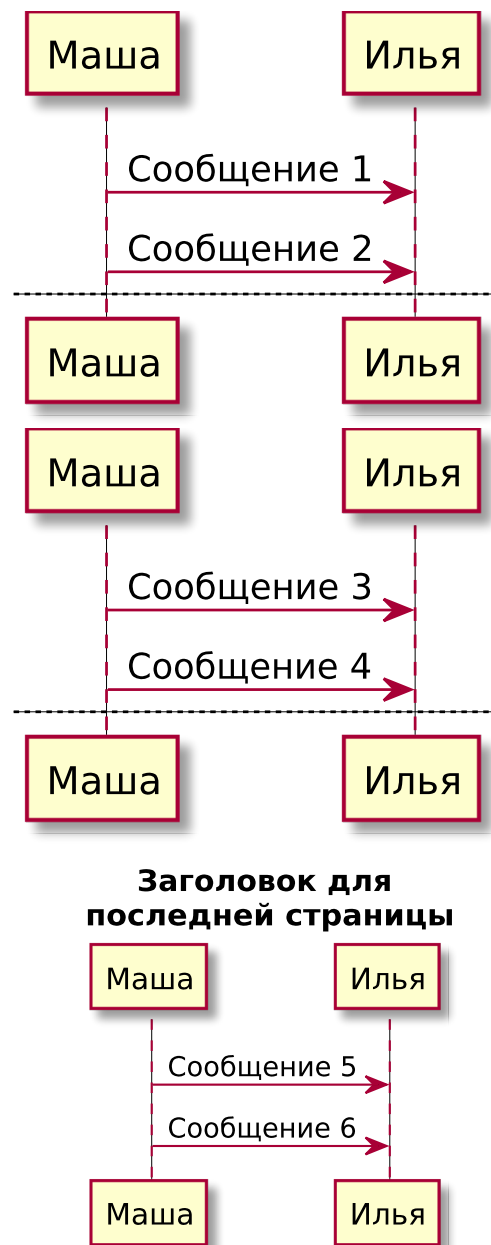
newpage

Маша -> Илья : Сообщение 3
Маша -> Илья : Сообщение 4

newpage Заголовок для \n последней страницы

Маша -> Илья : Сообщение 5
Маша -> Илья : Сообщение 6

@enduml
```



1.8 Группировка сообщений

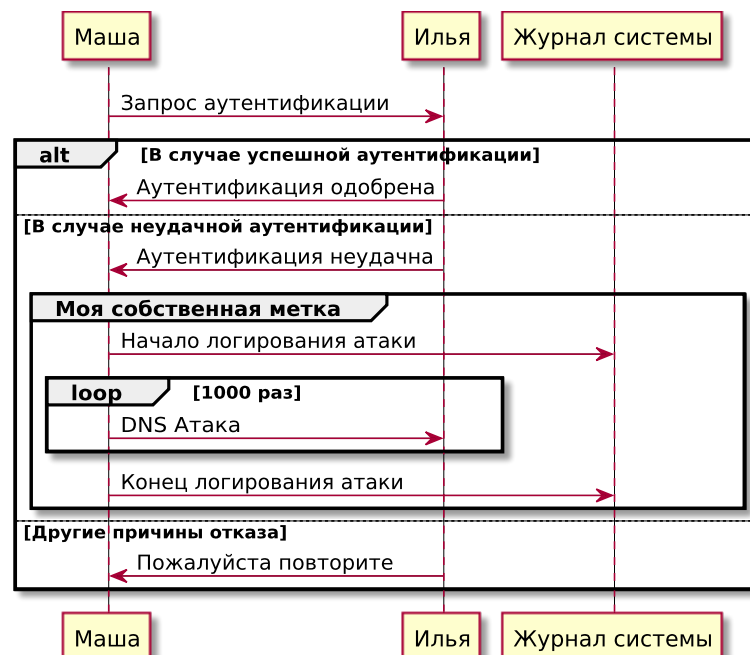
Для группировки сообщений используются следующие ключевые слова (группировке подлежит текст следующий за командой):

- alt/else
- opt
- loop
- par
- break

- critical
- group,

Это дает возможность добавления текста в заголовок. Ключевое слово «end» используется, чтобы завершить группировку. Обратите внимание на то, что существует возможность вложить одну группу в другую.

```
@startuml
Маша -> Илья : Запрос аутентификации
alt В случае успешной аутентификации
    Илья -> Маша : Аутентификация одобрена
else В случае неудачной аутентификации
    Илья -> Маша : Аутентификация неудачна
    group Моя собственная метка
        Маша -> "Журнал системы": Начало логирования атаки
        loop 1000 раз
            Маша -> Илья: DNS Атака
        end
        Маша -> "Журнал системы": Конец логирования атаки
    end
else Другие причины отказа
    Илья -> Маша : Пожалуйста повторите
end
@enduml
```



1.9 Примечания к сообщениям

Можно добавить примечания к сообщениям при помощи ключевых слов:

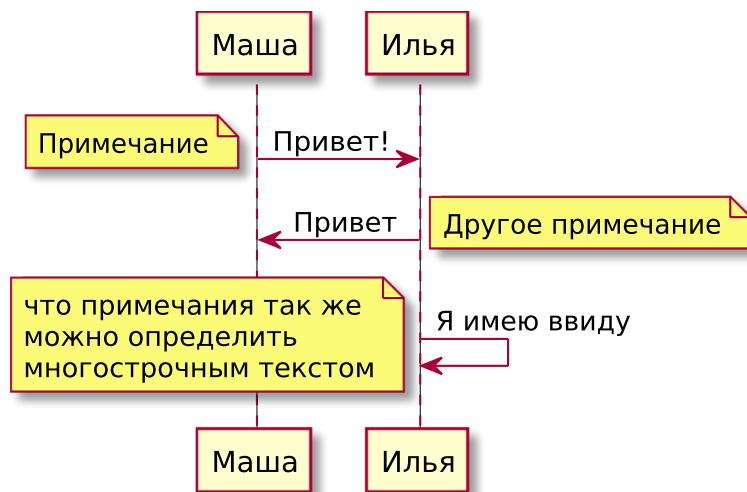
- «note left»,
- «note right».

Для многострочных примечаний используют ключевое слово «end note».

```
@startuml
Маша->Илья : Привет!
note left: Примечание

Илья->Маша : Привет
note right: Другое примечание

Илья->Илья : Я имею ввиду
note left
    что примечания так же
    можно определить
    многострочным текстом
end note
@enduml
```



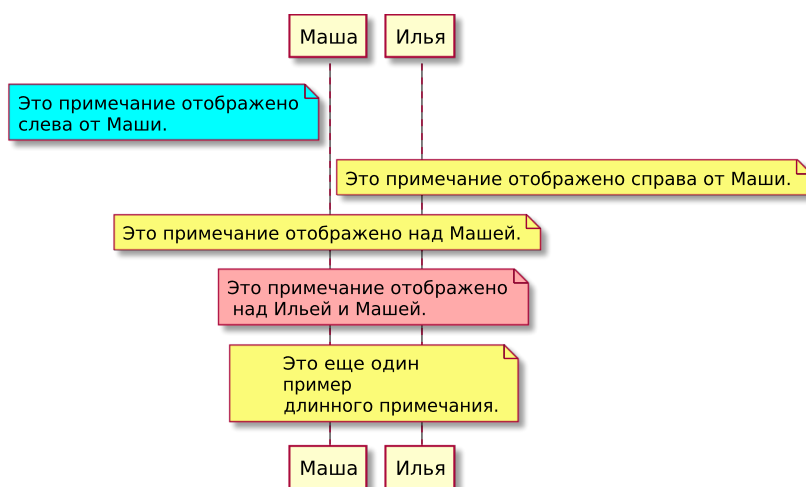
1.10 Другие примечания

Создать примечания так же можно при помощи следующих ключевых слов:

- «note left of»,
- «note right of»,
- «note over».

Для выделения примечаний можно изменять фон. При помощи ключевого слова «end note» можно создавать многострочные примечания.

```
@startuml
participant Маша
participant Илья
note left of Маша #aqua
    Это примечание отображено
    слева от Маши.
end note
note right of Маша: Это примечание отображено справа от Маши.
note over Маша: Это примечание отображено над Машей.
note over Маша, Илья #FFAAAA: Это примечание отображено\n над Ильей и Машей.
note over Илья, Маша
    Это еще один
    пример
    длинного примечания.
end note
@enduml
```



1.11 Форматирование с помощью HTML

Возможно использование некоторых html-тэгов:

- ``
- `<u>`
- `<i>`
- `<s>`, ``, `<strike>`
- `` или ``
- `<color:#AAAAAA>` или `<color:colorName>`
- `<size:nn>` для изменения размера шрифта
- `` или `<img:file>` : файл должен быть доступен в файловой системе

```
@startuml
participant Маша
participant "<b>Тот самый</b> Илья" as Илья
Маша -> Илья : <i>Отформатированное</i> сообщение
note right of Маша
    Это примечание <size:18>отображено</size>
    <u>слева от</u> Маши.
end note
note left of Илья
    Это примечание<color:#118888>отображено</color>
    <b><color:purple>слева от</color> <strike>Маши</strike> Ильи</b>.
end note
note over Маша, Илья
    Можно вставить картинку <img:picture.jpg>
end note
@endum
```



1.12 Разделитель

Для разделения схемы , используется « == » разделитель, для деления диаграммы на логические шаги.

@startuml

== Инициализация ==

Маша -> Илья: Запрос аутентификации

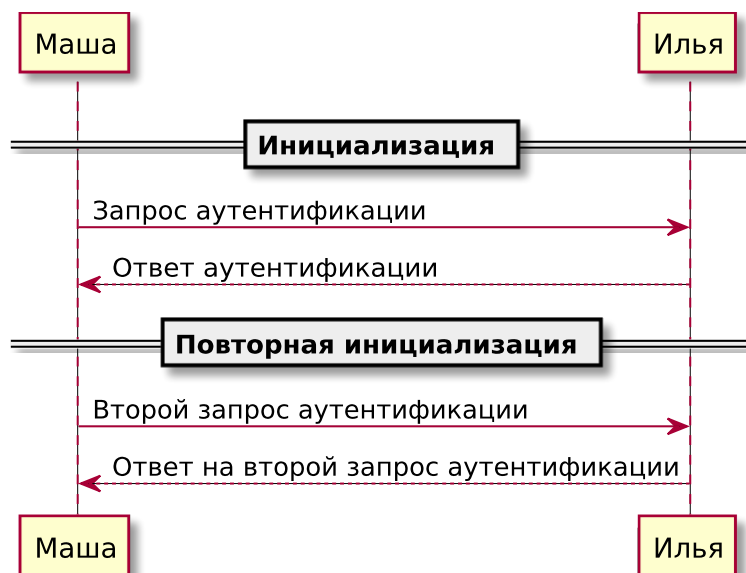
Илья --> Маша: Ответ аутентификации

== Повторная инициализация ==

Маша -> Илья: Второй запрос аутентификации

Маша <-- Илья: Ответ на второй запрос аутентификации

@enduml



1.13 Линии жизни. Активация и завершение

Команды `activate` и `deactivate` используются чтобы указать на участника активации. Когда участник активирован, появляется «линия жизни». Результат `activate` и `deactivate` применяется к сообщению указанному за командой. Команда `destroy` завершает «линию жизни» участника.

```
@startuml
participant Пользователь

Пользователь -> A: Выполнение
activate A

A -> B: << Создание запроса >>
activate B

B -> C: Выполнение
activate C
C --> B: Завершение
destroy C

B --> A: Запрос создан
deactivate B

A -> Пользователь: Завершено
deactivate A
@enduml
```



Можно использовать вложенные линии жизни, а так же указывать их цвет.


```
@startuml
participant Пользователь

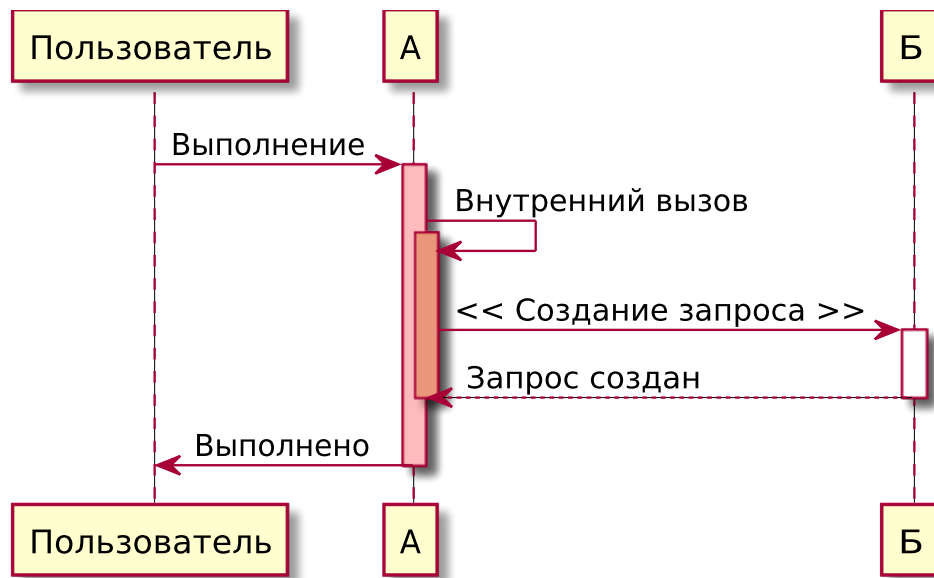
Пользователь -> A: Выполнение
activate A #FFBBBB

A -> A: Внутренний вызов
activate A #DarkSalmon

A -> Б: << Создание запроса >>
activate Б

Б --> A: Запрос создан
deactivate Б
deactivate A
A -> Пользователь: Выполнено
deactivate A

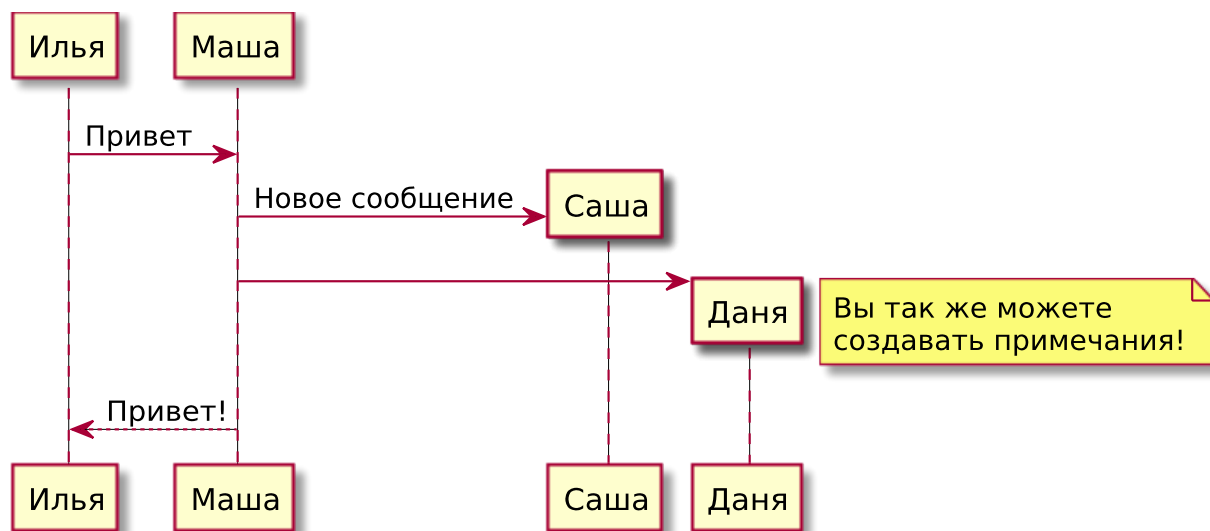
@enduml
```



1.14 Создание участников

При использовании ключевого слова «**create**» перед первым сообщением, вы подчеркиваете тот факт, что это сообщение фактически *создает* новый объект.

```
@startuml
Илья -> Маша : Привет
create "Саша"
Маша -> "Саша" : Новое сообщение
create "Даня"
Маша -> "Даня"
note right : Вы так же можете \n создавать примечания!
Маша --> Илья : Привет!
@enduml
```



1.15 Входящие и исходящие сообщения

Вы можете использовать входящие или исходящие стрелки, если хотите сфокусироваться на какой-либо части диаграммы.

Используйте квадратные скобки для указания на левую «[» или правую «]» сторону схемы.

```
@startuml
```

```
[-> A: Выполнение
```

```
activate A
```

```
A -> A: Внутренний вызов
```

```
activate A
```

```
A ->] : << Создание запроса >>
```

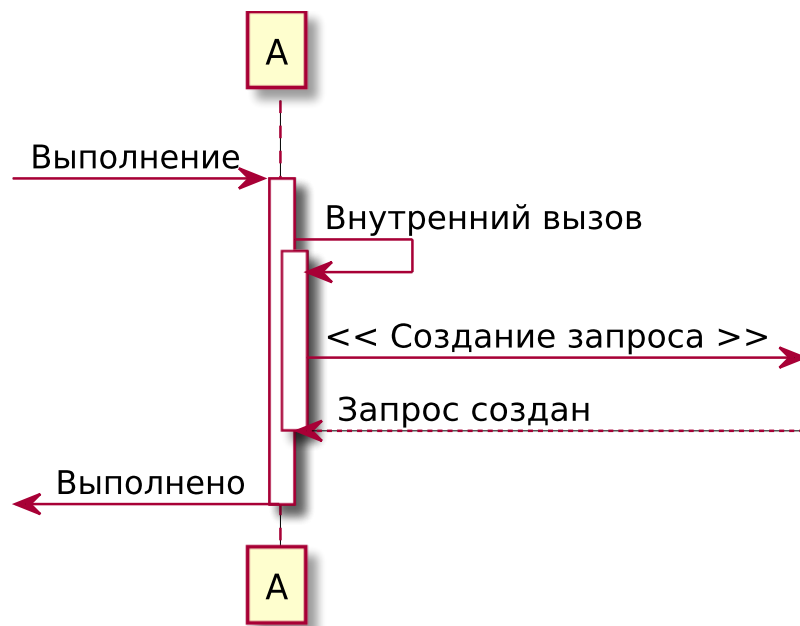
```
A<--] : Запрос создан
```

```
deactivate A
```

```
[<- A: Выполнено
```

```
deactivate A
```

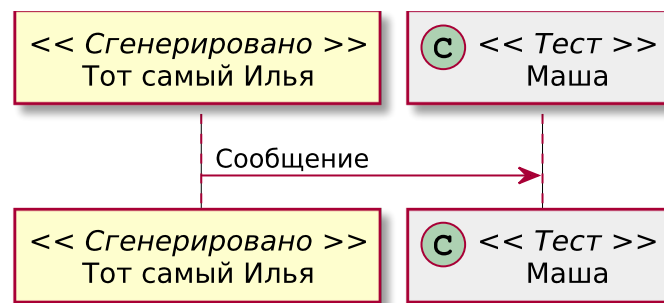
```
@enduml
```



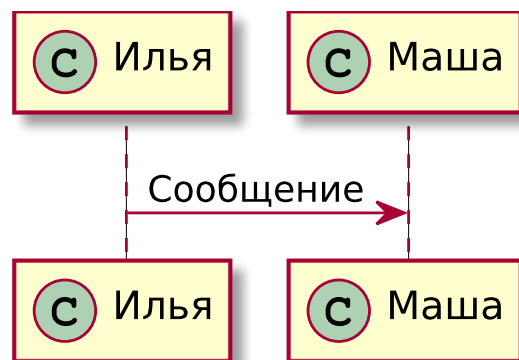
1.16 Стереотипы и иконки

Стереотип отображается как имя, заключенное в кавычки «<<» и «>>». В дополнение он может быть обозначен соответствующей иконкой при помощи синтаксиса «(X, color)».

```
@startuml
participant "Тот самый Илья" as Илья << Сгенерировано >>
participant Маша << (C,#ADD1B2) Тест >> #EEEEEE
Илья->Маша: Сообщение
@enduml
```



```
@startuml
participant Илья << (C,#ADD1B2) >>
participant Маша << (C,#ADD1B2) >>
Илья->Маша: Сообщение
@enduml
```



1.17 Больше информации о заголовках

Вы можете использовать некоторые HTML-тэги в заголовке.

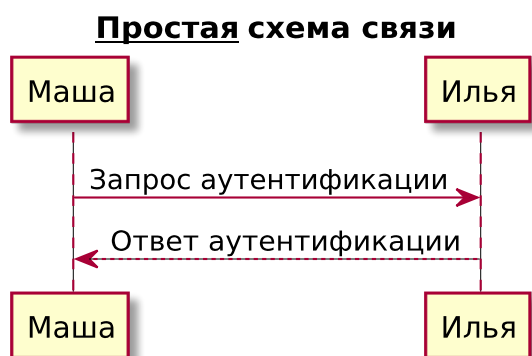
```
@startuml
```

```
title <u>Простая</u> схема связи
```

```
Маша -> Илья: Запрос аутентификации
```

```
Илья --> Маша: Ответ аутентификации
```

```
@enduml
```



Вы можете добавить новую строку, используя «\n» в описании заголовка.

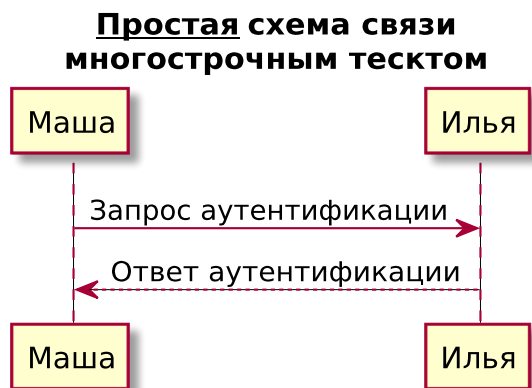
```
@startuml
```

```
title <u>Простая</u> схема связи \n многострочным текстом
```

```
Маша -> Илья: Запрос аутентификации
```

```
Илья --> Маша: Ответ аутентификации
```

```
@enduml
```



Вы можете также разместить заголовок на нескольких строках, используя ключевые слова «title» и «end title».

```
@startuml
```

```
title
```

```
<u>Простая</u> схема связи
```

```
<i>многострочным</i> текстом с использованием <font color=red>html</font>.
```


```
Так же можно вставлять картинку <img src=picture.jpg>
```

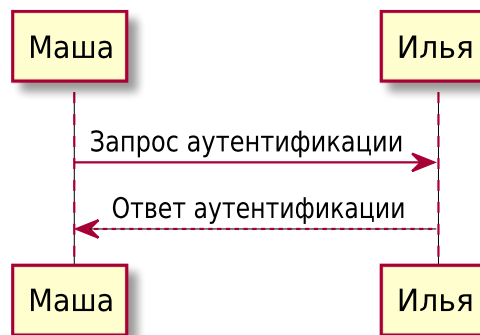
```
end title
```

```
Маша -> Илья: Запрос аутентификации
```

```
Илья --> Маша: Ответ аутентификации
```

```
@enduml
```

Простая схема связи
многострочным текстом с использованием html.
Так же можно вставлять картинку 



1.18 Выделение участников

Возможно протянуть цветное поле на нескольких участников, используя команды «**box**» и «**end box**».

Вы можете добавить дополнительный заголовок или дополнительный цвет фона после ключевого слова «**box**».

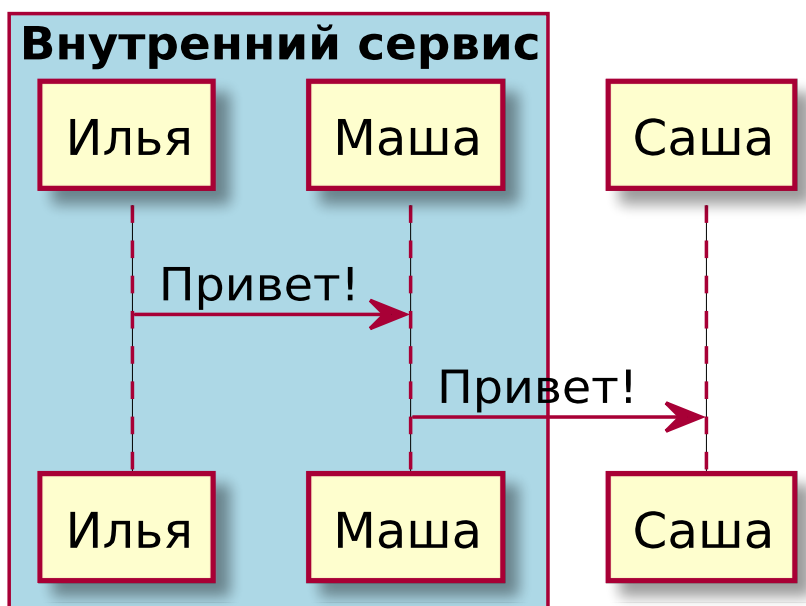
```
@startuml
```

```
box "Внутренний сервис" #LightBlue
participant Илья
participant Маша
end box
participant Саша
```

```
Илья -> Маша : Привет!
```

```
Маша -> Саша : Привет!
```

```
@enduml
```



1.19 Удаление нижнего колонтитула

Вы можете использовать ключевое слово «hide footer», чтобы удалить нижний колонтитул диаграммы.

```
@startuml
```

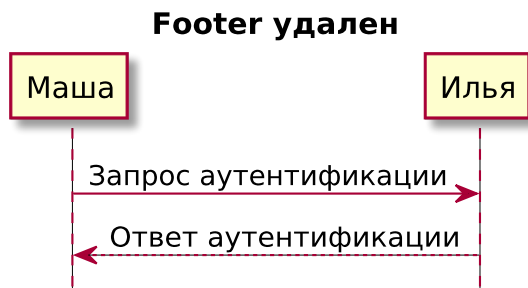
```
hide footer
```

```
title Footer удален
```

```
Маша -> Илья: Запрос аутентификации
```

```
Илья --> Маша: Ответ аутентификации
```

```
@enduml
```



1.20 Команда «skinparam»

Вы можете использовать команду «skinparam», чтобы изменить цвета и шрифты для диаграммы. Также возможно использование этой команды для:

- Определения диаграммы, как и для других любых команд,
- Импортированных файлов,
- Файла конфигурации, вызванного через командную строку.

```
@startuml
```

```
skinparam backgroundColor #EEEEBD
```

```
skinparam sequenceArrowColor DeepSkyBlue
skinparam sequenceParticipantBorderColor DeepSkyBlue
skinparam sequenceActorBorderColor DeepSkyBlue
skinparam sequenceLifeLineBorderColor blue
```

```
skinparam sequenceParticipantBackgroundColor DodgerBlue
skinparam sequenceParticipantFontName Impact
skinparam sequenceParticipantFontSize 17
skinparam sequenceParticipantFontColor #A9DCDF
```

```
skinparam sequenceActorBackgroundColor aqua
skinparam sequenceActorFontColor DeepSkyBlue
skinparam sequenceActorFontSize 17
skinparam sequenceActorFontName Aapex
```

```
skinparam sequenceLifeLineBackgroundColor #A9DCDF
```

```
actor Пользователь
participant "Первый класс" as A
participant "Второй класс" as B
participant "Последний класс" as C
```

```
Пользователь -> A: Выполнение
activate A
```

```
A -> B: Создание запроса
activate B
```

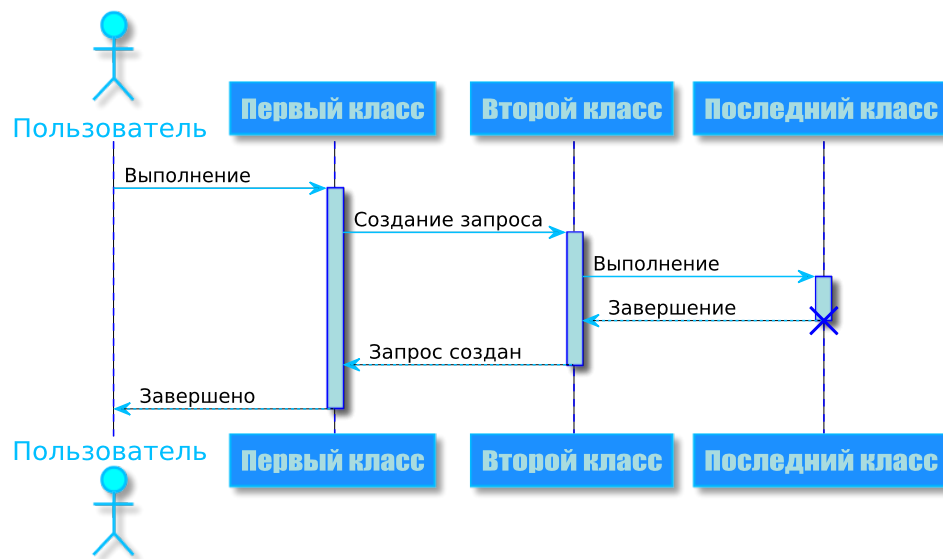
```
B -> C: Выполнение
activate C
```

```
C --> B: Завершение
destroy C
```

Б --> А: Запрос создан
deactivate Б

А --> Пользователь: Завершено
deactivate А

@endum1



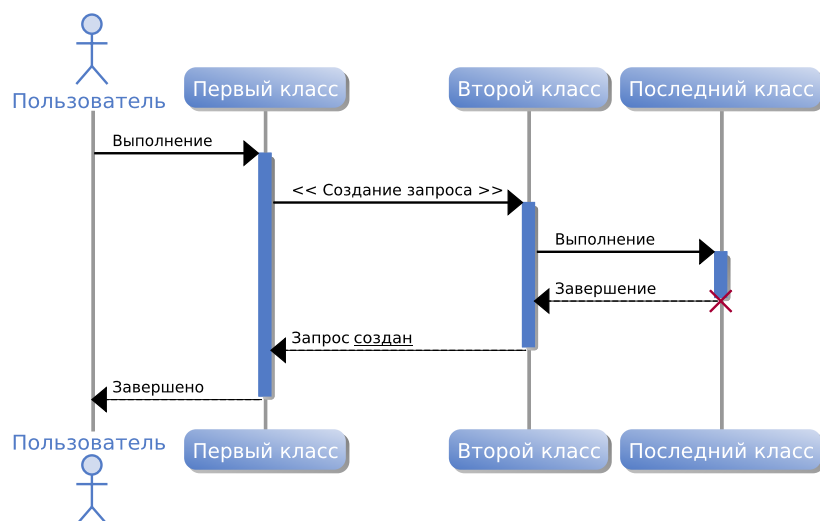
1.21 Команда «skin»

Используйте ключевое слово «**skin**», чтобы изменить вид сгенерированной схемы. На сегодняшний день есть только два доступных вида (*Rose*, который является значением по умолчанию и *BlueModern*), но можно создать ваш собственный «**skin**».

```
@startuml
skin BlueModern

actor Пользователь
participant "Первый класс" as A
participant "Второй класс" as B
participant "Последний класс" as C

Пользователь -> A: Выполнение
activate A
A -> B: << Создание запроса >>
activate B
B -> C: Выполнение
activate C
C --> B: Завершение
destroy C
B --> A: Запрос <u>создан</u>
deactivate B
A --> Пользователь: Завершено
deactivate A
@enduml
```



2. Диаграмма прецедентов

2.1 Прецеденты

Прецеденты включаются круглыми скобками (образуют овал). Также возможно обозначение прецедента командой «**usecase**». Наименование прецедента возможно посредством команды «**as**». Имя прецедента будет использоваться и дальше, при определении отношений между экторами.

```
@startuml
(Прецедент)
(Другой прецедент) as (П2)
usecase П3
usecase (Последний \n прецедент) as П4
@enduml
```



2.2 Экторы

Для определения эктора используется две точки или команда «actor», чтобы задать имя эктору – «as», имя понадобится для описания отношений между экторами. Следует заметить, что эктор является опцией и не обязателен для использования.

```
@startuml
:Актер:
:Другой \n актер: as N2
actor N3
actor :Последний актер: as N4
@enduml
```



2.3 Примеры

Чтобы соединить эктора и прецедент используется прерывистая стрелка «- ->». Чем больше штрихов «-» в стрелке, тем длинее нарисоваться стрелка. Вы можете указать имя стрелки, путем добавления символа «:» в определении стрелки. В этом примере вы можете видеть, что *Пользователь* не был определен заранее.

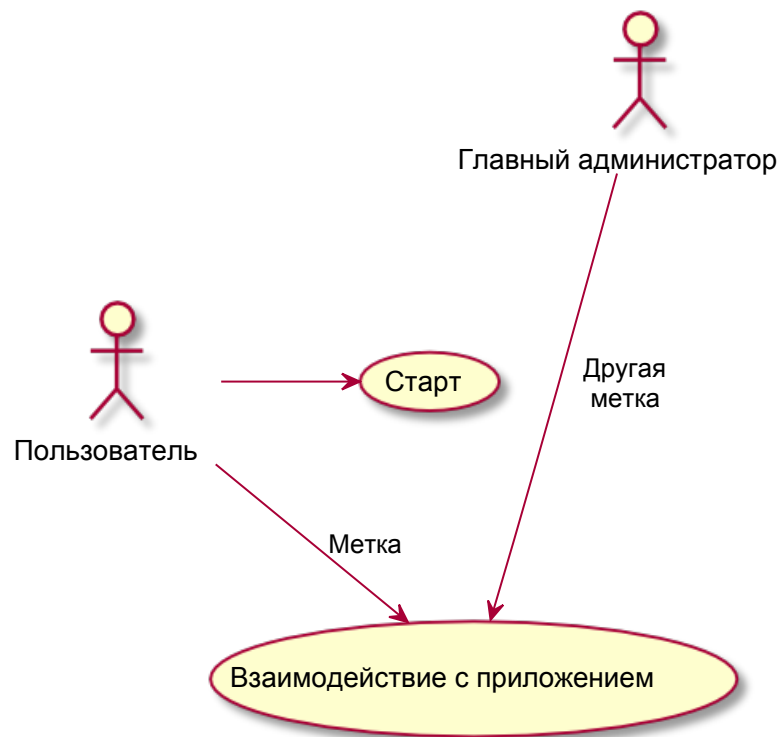
```
@startuml
```

```
Пользователь -> (Старт)
```

```
Пользователь --> (Взаимодействие с приложением) : Метка
```

```
:Главный администратор: ---> (Взаимодействие с приложением) : Другая \n метка
```

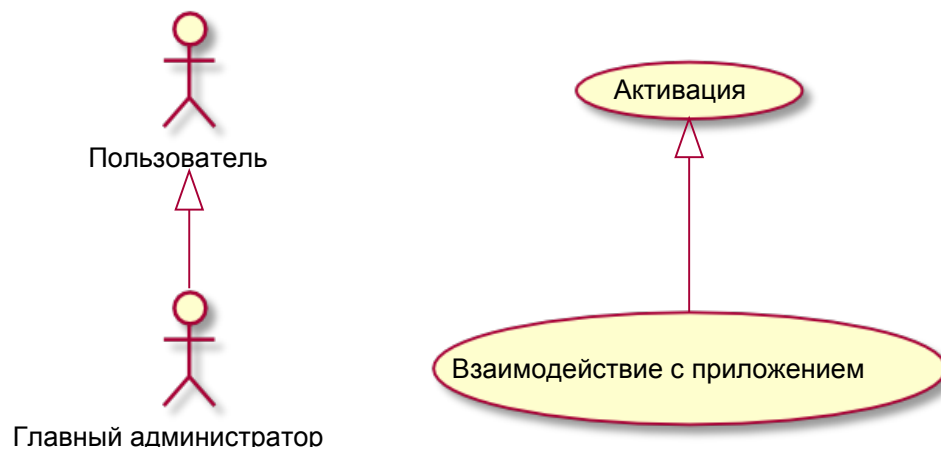
```
@enduml
```



2.4 Наследование или обобщение

Чтобы обозначить наследование одним прецедентом(эктором) другого применяется комбинация «<|--». После запуска команды вы обнаружите появление вертикально направленной стрелки.

```
@startuml
:Главный администратор: as Администратор
(Взаимодействие с приложением) as (Действие)
Пользователь <|-- Администратор
(Активация) <|-- (Действие)
@enduml
```



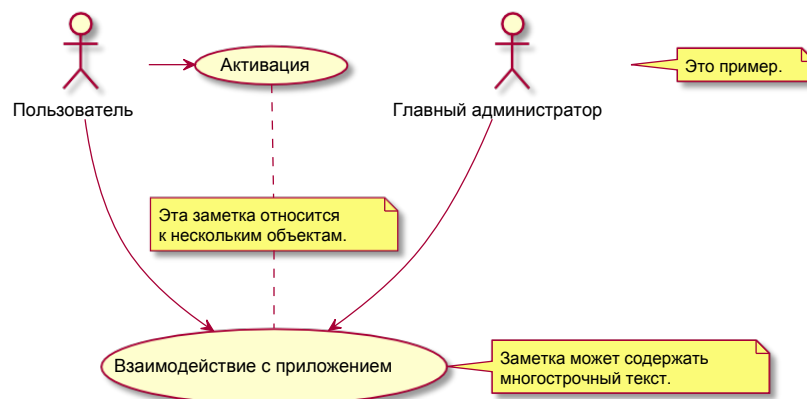
2.5 Примечания

Для добавления примечаний к одиночному объекту вы можете использовать такие ключевые слова как:

- note left of,
- note right of,
- note top of,
- note bottom of

Также заметки могут быть описаны отдельно с помощью ключевого слова «**note**» и присоединены к объекту комбинацией «. .».

```
@startuml
:Главный администратор: as Администратор
(Взаимодействие с приложением) as (Действие)
Пользователь -> (Активация)
Пользователь --> (Действие)
Администратор ---> (Действие)
note right of Администратор : Это пример.
note right of (Действие)
  Заметка может содержать
  многострочный текст.
end note
note "Эта заметка относится \n к нескольким объектам." as Зам2
(Активация) .. Зам2
Зам2 .. (Действие)
@enduml
```



2.6 Стереотипы

Используя комбинацию символов « << >> », вы можете добавить стереотип перед эктором или прецедентом.

@startuml

Пользователь << Человек >>

:Главная база данных: as MySql << Приложение >>

(Активация) << Один раз >>

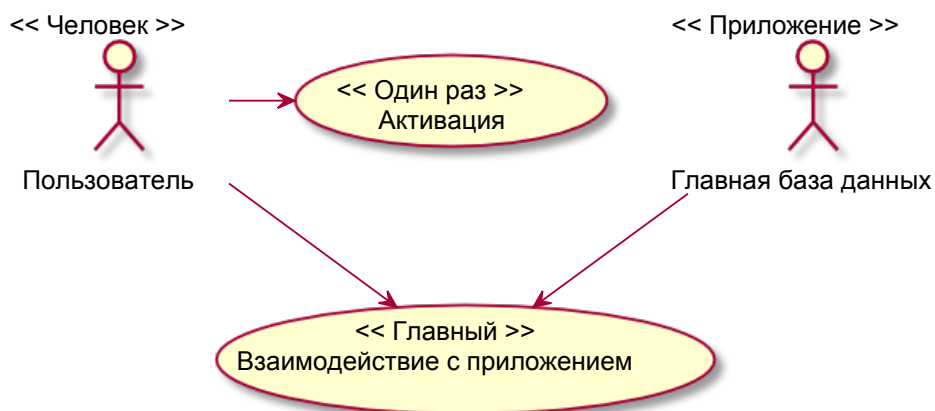
(Взаимодействие с приложением) as (Действие) << Главный >>

Пользователь -> (Активация)

Пользователь --> (Действие)

MySql --> (Действие)

@enduml

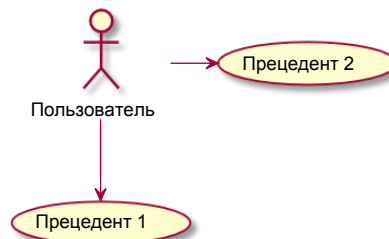


2.7 Изменение направления стрелок

По умолчанию, соединяющие стрелки между классами вертикально направлены и состоят из двух черточек «- -». Также существует возможность изменить вертикальное направление стрелки на горизонтальное, используя одну черточку «-» (или точку).

```
:Пользователь: --> (Прецедент 1)
```

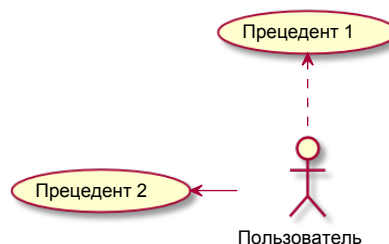
```
:Пользователь: -> (Прецедент 2)
```



Также можно менять направление стрелки непосредственно в коде:

```
(Прецедент 1) <.. :Пользователь:
```

```
(Прецедент 2) <- :Пользователь:
```



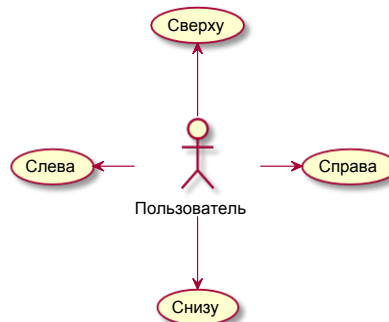
Также можно менять направление стрелок путем добавления ключевых слов «left», «right», «up» или «down», поместив внутри тела стрелки в коде.

```
:Пользователь: -left-> (Слева)
```

```
:Пользователь: -right-> (Справа)
```

```
:Пользователь: -up-> (Сверху)
```

```
:Пользователь: -down-> (Снизу)
```



Вы можете сделать стрелки короче, используя в коде только первую букву команды, например, «-d-» вместо «-down-» или первые две буквы «-do-». Обратите внимание на то, что вы не должны перегрузить функциональную часть: *Graph Viz* не требует доработок и дает прекрасный результат.

2.8 Название диаграммы

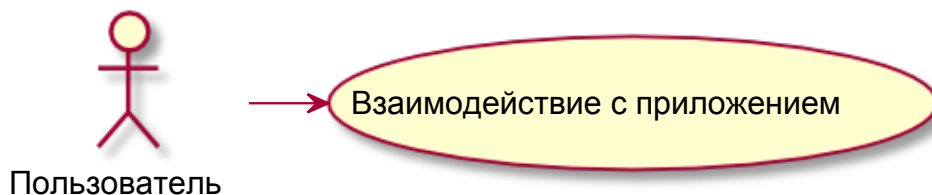
Чтобы дать имя диаграмме используется ключевое слово «**title**». Для длинного названия диаграммы используется сразу две команды «**title**» и «**end title**», как и в случае с диаграммой последовательности.

```
@startuml
```

```
title Пример <b>диаграммы прецедентов </b>\n  с одним актером  
usecase (Взаимодействие с приложением) as (Действие)  
Пользователь -> (Действие)
```

```
@enduml
```

Пример диаграммы прецедентов
с одним актером



2.9 Направление диаграммы

По умолчанию построение диаграммы начинается сверху вниз.

```
@startuml
```

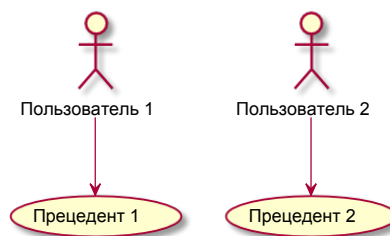
```
'По умолчанию направление сверху вниз
```

```
top to bottom direction
```

```
:Пользователь 1: --> (Прецедент 1)
```

```
:Пользователь 2: --> (Прецедент 2)
```

```
@enduml
```



Вы можете поменять порядок построения путем использования команды «left to right direction». Результат выглядит намного лучше, чем при построении сверху вниз.

```
@startuml
```

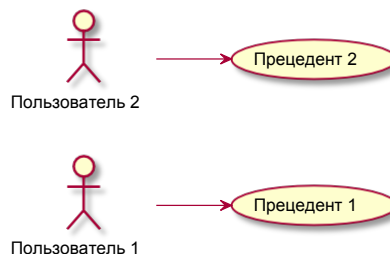
```
'Направление слева направо
```

```
left to right direction
```

```
:Пользователь 1: --> (Прецедент 1)
```

```
:Пользователь 2: --> (Прецедент 2)
```

```
@enduml
```



2.10 Команда «skinparam»

Вы можете использовать команду «skinparam», чтобы изменить цвета и шрифты для диаграммы. Также возможно использование этой команды для:

- Определения диаграммы, как и для других любых команд,
- Импортированных файлов,
- Файла конфигурации, вызванного через командную строку.

```
@startuml
```

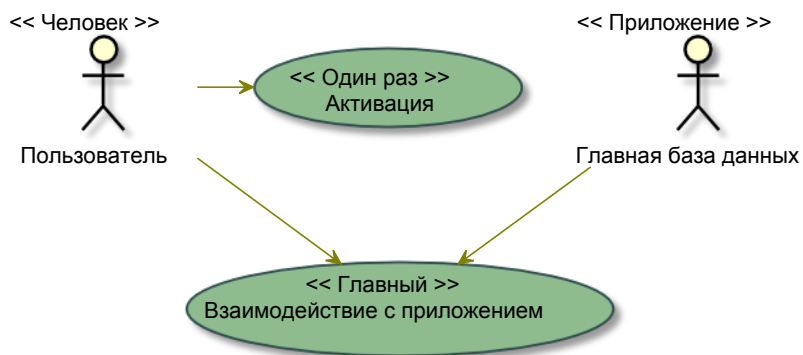
```
skinparam usecaseBackgroundColor DarkSeaGreen
skinparam usecaseArrowColor Olive
skinparam actorBorderColor black
skinparam usecaseBorderColor DarkSlateGray
skinparam usecaseActorFontName Courier
```

```
Пользователь << Человек >>
:Главная база данных: as MySql << Приложение >>
(Активация) << Один раз >>
(Взаимодействие с приложением) as (Действие) << Главный >>
```

```
Пользователь -> (Активация)
Пользователь --> (Действие)
```

```
MySql --> (Действие)
```



```
@enduml
```



3. Диаграмма классов

3.1 Связи между классами

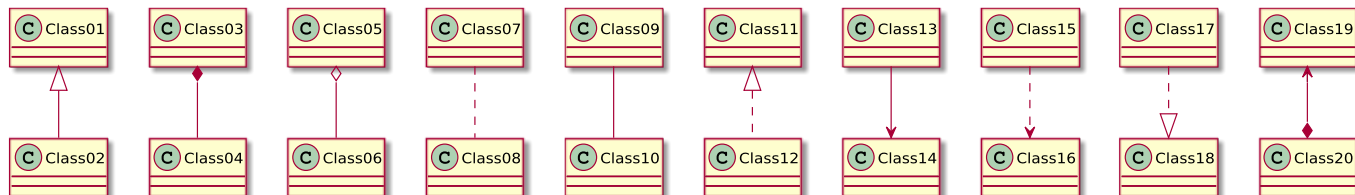
Связи между классами определяются при помощи следующих символов:

Обобщение	< --	
Композиция	*--	
Агрегация	o--	

При замене «--» на «..» будет отображаться пунктирная линия.

Следуя этим правилам можно нарисовать следующее:

```
@startuml
  Class01 <|-- Class02
  Class03 *-- Class04
  Class05 o-- Class06
  Class07 .. Class08
  Class09 -- Class10
  Class11 <|.. Class12
  Class13 --> Class14
  Class15 ..> Class16
  Class17 ..|> Class18
  Class19 <--* Class20
@enduml
```

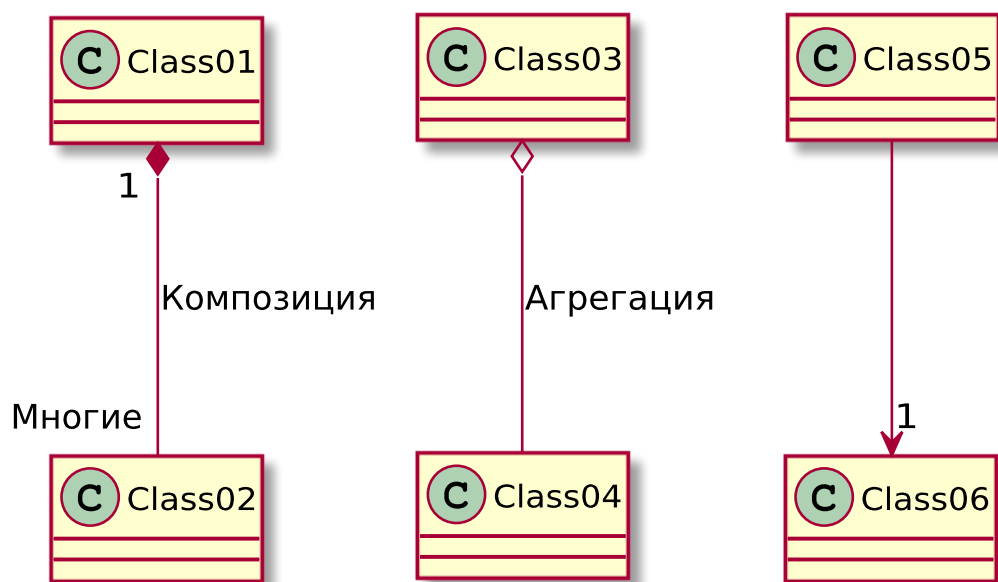


3.2 Пометки на связях

Для того чтобы добавить пометку к связи, используется символ «:», сообщение следующее за командой станет пометкой.

Для указания кратности, используются двойные кавычки «» с каждой стороны связи.

```
@startuml
  Class01 "1" *-- "многие" Class02 : Композиция
  Class03 o-- Class04 : Агрегация
  Class05 --> "1" Class06
@enduml
```

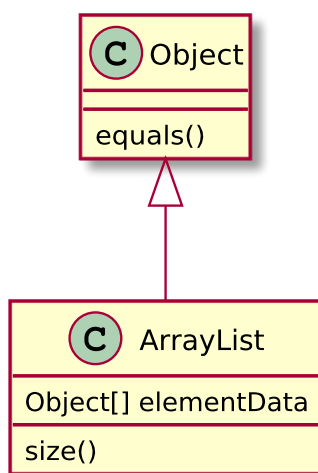


3.3 Добавление методов

При объявлении методов и полей используется символ «:» последующий текст станет именем метода или поля.

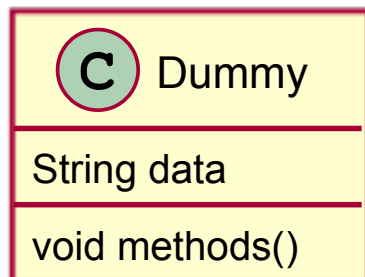
Система проверяет наличие круглых скобок, для определения метода или поля

```
@startuml
Object <|-- ArrayList
Object : equals()
ArrayList : Object[] elementData
ArrayList : size()
@enduml
```



Так же возможно выполнять группировку методов и полей между фигурными скобками «{}»

```
@startuml
class Dummy {
String data
void methods()
}
@enduml
```

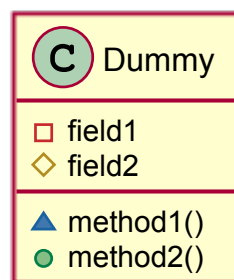


3.4 Определение видимости

Когда вы определяете метод или поле, можно использовать специальные символы для указания видимости соответствующих объектов

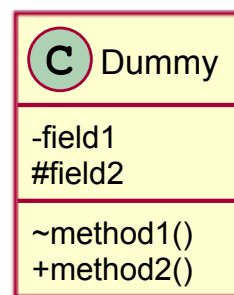
-	□	■	private	личный
#	◇	◆	protected	защищенный
~	△	▲	package private	публичный в пределах пакета
+	○	●	public	публичный

```
@startuml
class Dummy {
    -field1
    #field2
    ~method1()
    +method2()
}
@enduml
```



Можно отключить отображение иконок при изменении значения параметра «classAttributeIconSize» на 0 в настройках skinparam:

```
@startuml
skinparam classAttributeIconSize 0
class Dummy {
    -field1
    #field2
    ?method1()
    +method2() }
@enduml
```



3.5 Заметки и стереотипы

Стереотипы определяются при помощи ключевого слова «class», «<<» и «>>».

Так же, заметки определяются при помощи следующих команд «note left of», «note right of», «note top of», «note bottom of».

Заметку можно объявить отдельно, при помощи команды «note», что свяжет её с другим объектом при помощи символа «..».

```
@startuml
```

```
class Object << general >>
```

```
Object <|--- ArrayList
```

```
note top of Object : В java, каждый класс \n расширяет базовый класс.
```

```
note "Это отдельная заметка" as N1
```

```
note "Эта заметка соединена \n с несколькими объектами." as N2
```

```
Object .. N2
```

```
N2 .. ArrayList
```

```
@enduml
```



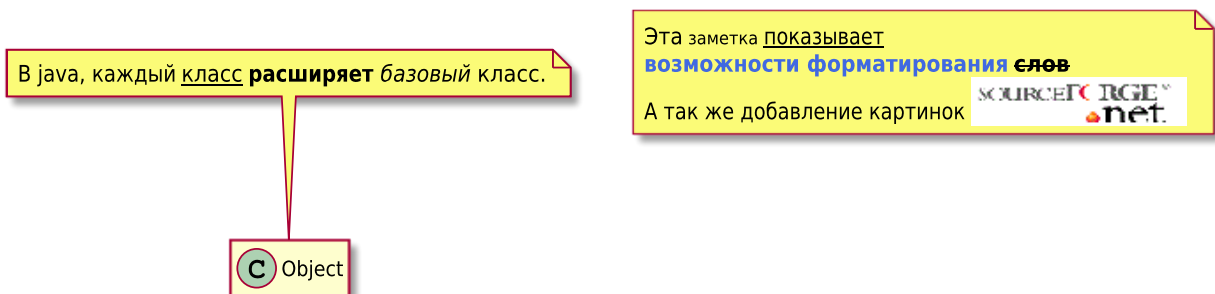
3.6 Дополнительно о заметках

Существует поддержка некоторых html тэгов, таких как:

- ``
- `<u>`
- `<i>`
- `<s>`, ``, `<strike>`
- `` или ``
- `<color:#AAAAAA>` или `<color:colorName>`
- `<size:nn>` для изменения размера шрифта
- `` или `<img:file>` : файл должен быть доступен в файловой системе

Так же есть возможность разбить заметку на несколько строк

```
@startuml
note top of Object
В java, каждый <u>класс</u>
<b>расширяет </b>
<i>родительский </i> класс.
end note
note as N1
Эта <size:10>заметка </size > просто <u> показывает </u>
<b><color:royalBlue >возможности форматирования </color >
<s>заметок</s>
А так же добавление картинок <img:sourceforge.jpg >
end note
@enduml
```



3.7 Абстрактные классы и интерфейсы

Вы можете объявлять классы абстрактными при помощи ключевых слов «**abstract**», «**abstract class**». Классы на печати отображаются курсивом

Так же есть поддержка ключевых слов «**interface**» и «**enum**».

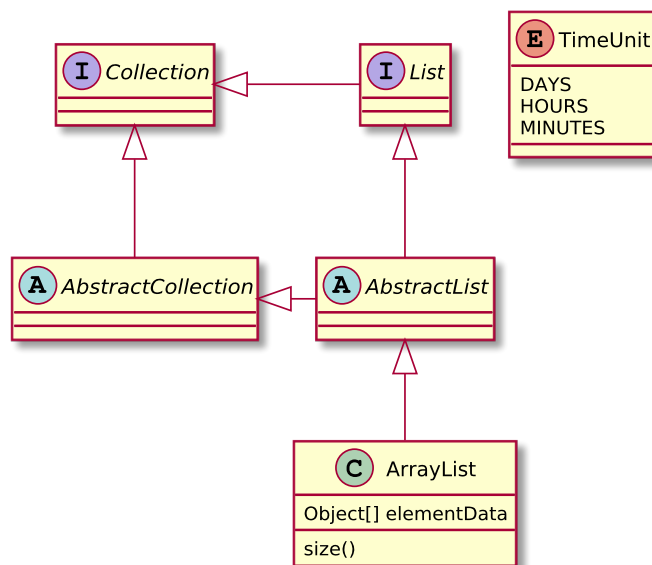
```
@startuml
    abstract class AbstractList
    abstract AbstractCollection
    interface List
    interface Collection

    List <|-- AbstractList
    Collection <|-- AbstractCollection

    Collection <|-- List
    AbstractCollection <|-- AbstractList
    AbstractList <|-- ArrayList

    ArrayList : Object[] elementData
    ArrayList : size()

    enum TimeUnit
        TimeUnit : DAYS
        TimeUnit : HOURS
        TimeUnit : MINUTES
    @enduml
```

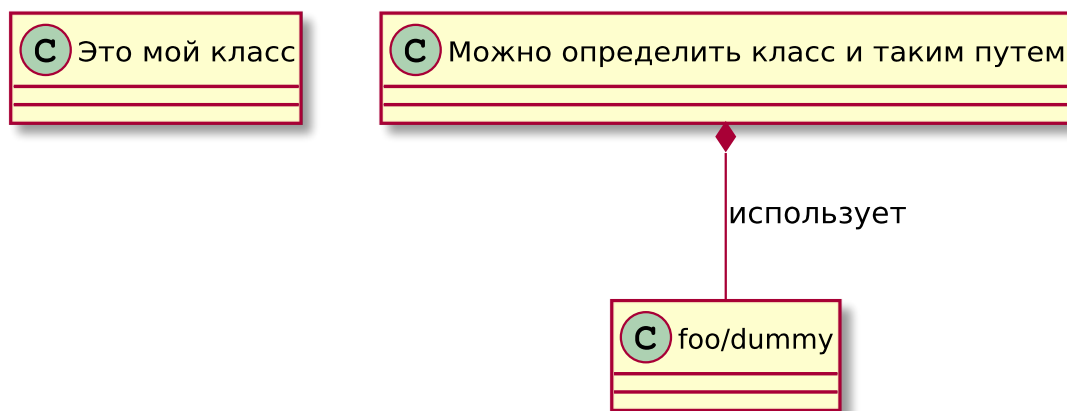


3.8 Использование спецсимволов

При желании можно использовать спецсимволы в классах, при помощи:

- Использования ключевого слова «**as**» при определении класса
- Постановки кавычек «' '» вокруг имени класса

```
@startuml
class "Это мой класс" as class1
class class2 as "А можно определить класс, и таким путем"
class2 *-- "Да/и так тоже " : использует
@enduml
```



3.9 Скрытые атрибуты и методы

В PlantUML есть поддержка параметризации отображения классов при помощи команд «hide/show».

Базовая команда выглядит так: «hide empty members». Эта команда скрывает атрибуты и методы, если они пустые.

Вместо «empty members», можно использовать следующие ключевые слова:

- «empty fields» или «empty attributes» для пустых полей,
- «empty methods» для пустых методов,
- «fields» или «attributes» для сокрытия полей, даже если они описаны,
- «methods» для сокрытия методов, даже если они описаны,
- «members» для сокрытия полей и методов, даже если они описаны,
- «circle» для специальных символов в заголовке класса,
- «stereotype» для стереотипов.

После чего, вы указываете после ключевых слов «hide» или «show» параметры:

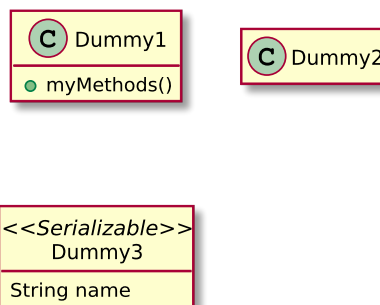
- «class» для всех классов,
- «interface» для всех интерфейсов,
- «enum» для всех перечислений,
- «<<foo1>>» для классов со стереотипами *foo1*,
- для конкретных классов.

Можно использовать несколько команд «show/hide», для определения правил и исключений.

```
@startuml
```

```
class Dummy1 {
    +myMethods()
}
class Dummy2 {
    +hiddenMethod()
}
class Dummy3 <<Serializable>> {
    String name
}
hide members
hide <<Serializable>> circle
show Dummy1 method
show <<Serializable>> fields
```

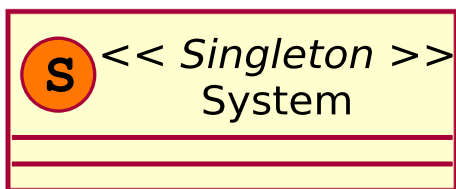
```
@enduml
```



3.10 Специальные значки

Обычно специальные значки (C, I, E, A) используются для отображения классов, интерфейсов, перечислений и абстрактных классов. Но вы можете определить свой значок для класса во время определения стереотипа, а так же указать специальный символ либо цвет.

```
@startuml
class System << (S,#FF7700) Singleton >>
class Date << (D,orchid) >>
@enduml
```



3.11 Пакеты

Вы можете определять пакеты при помощи команды «`package`» и дополнительного указания цвета заднего фона (при помощи html кода цвета). Когда вы объявляете классы, они автоматически становятся членами последнего используемого пакета. Для завершения объявления пакета используется команда «`end package`». Имеется поддержка фигурных скобок «`{}`».

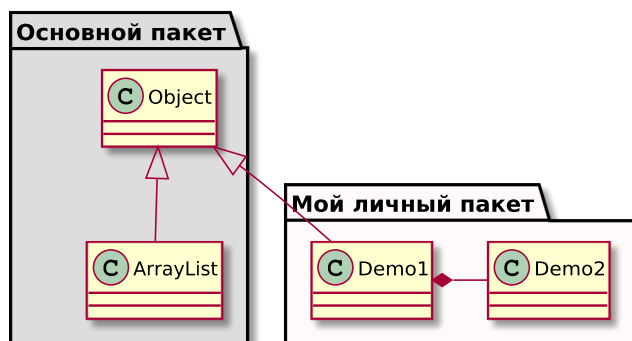
Осуществленна поддержка вложенности пакетов.

```
@startuml
```

```
package "Основной пакет" #DDDDDD {
  Object <|-- ArrayList
}
```

```
package Мой личный пакет #Snow
  Object <|-- Demo1
  Demo1 *- Demo2
end package
```

```
@enduml
```



Можно указать на связь между пакетами, например так:

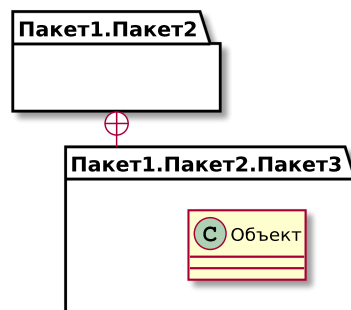
```
@startuml
```

```
package пакет1.пакет2{
}
```

```
package пакет1.пакет2.пакет3 {
  class Object
}
```

```
пакет1.пакет2 +-- пакет1.пакет2.пакет3
```

```
@enduml
```



3.12 Пространство имен

В пакете, имя класса это уникальный идентификатор. Таким образом у вас не может быть одинаковых имен классов в разных пакетах. В таком случае вы должны использовать пространства имен вместо пакетов.

Вы можете обратиться к классам из других пространств имен, путем указания полного имени. Классы из стандартного пространства имен определяются путем добавления стартовой точки.

Обратите внимание на то, что вы не должны создавать пространство имен: полное, правильное наименование класса автоматически будет переведено в пространство имен.

@startuml

```
class BaseClass
```

```
namespace net.dummy #DDDDDD
```

```
  .BaseClass <|-- Person
```

```
  Meeting o-- Person
```

```
  .BaseClass <|-- Meeting
```

```
end namespace
```

```
namespace net.foo {
```

```
  net.dummy.Person <|-- Person
```

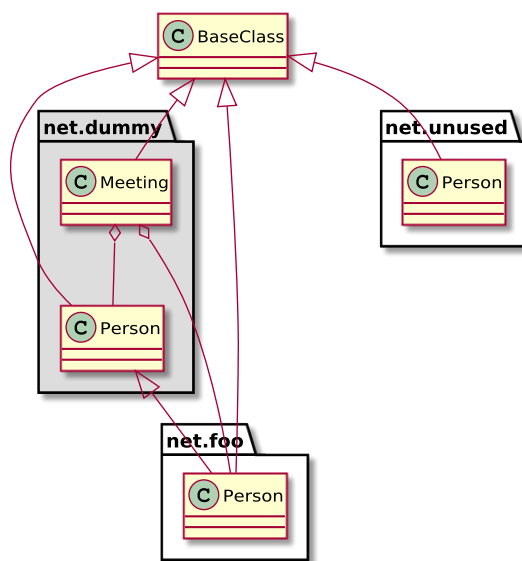
```
  .BaseClass <|-- Person
```

```
  net.dummy.Meeting o-- Person
```

```
}
```

```
BaseClass <|-- net.unused.Person
```

@enduml

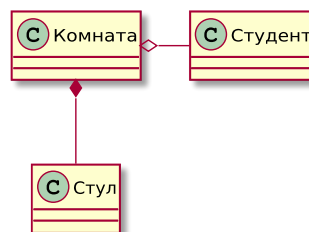


3.13 Изменение направления стрелок

По умолчанию, связи между двумя классами определяются двумя тире «- -». Так же возможно отображение в виде одного тире и специального символа (например *, o).

Комната o- Студент

Комната *-- Стул



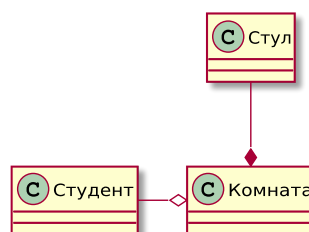
Изменение направления стрелок происходит путем перестановки объектов в связи:

```
@startuml
```

```
Студент -o Комната
```

```
Стул --* Комната
```

```
@enduml
```



Так же возможна смена направления стрелок путем указания ключевых слов «left», «right», «top», «down» внутри стрелок, например:

```
@startuml
```

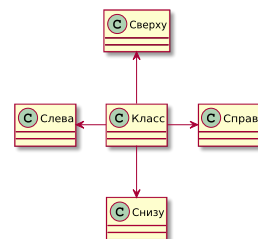
```
Класс -left -> КлассСлева
```

```
Класс -right -> КлассСправа
```

```
Класс -up-> КлассСверху
```

```
Класс -down -> КлассСнизу
```

```
@enduml
```



Для простоты набора, во время указания направления стрелки, можно писать не целое слово, а его часть например «-d-» или «-do-» вместо «down»

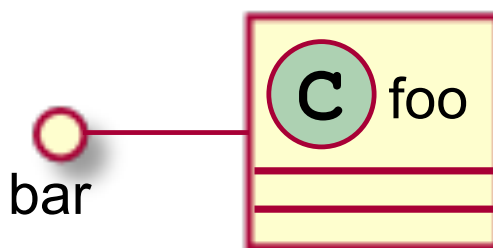
Пожалуйста, не злоупотребляйте ручной настройкой, так как *GraphViz* (пакет утилит по автоматической визуализации графов) в большинстве случаев показывает хорошие результаты.

3.14 Интерфейс «lolipop»

Вы можете использовать интерфейс «lolipop» в диаграмме классов, используя следующий синтаксис:

- «Класс ()- Интерфейс»
- «Класс ()- - Интерфейс»
- «Интерфейс -() Класс»

```
@startuml
class Класс
Класс ()- Интерфейс
@enduml
```

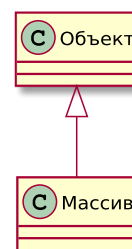


3.15 Заголовок диаграммы

Для постановки текста в заголовок диаграммы используется ключевое слово «**title**». Для многострочных заголовков соответственно «**title**» и «**end title**», как и в диаграммах последовательностей.

```
@startuml
title Просто <b>пример </b>\n заголовка
Объект <|-- СписокМассива
@enduml
```

Простой **пример**
заголовка

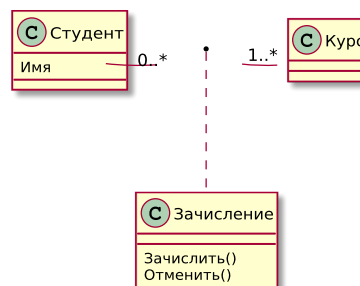


3.16 Ассоциация классов

Вы можете указать ассоциацию классов, после определения связей между двумя классами, как на этом примере:

```
@startuml
Студент : Имя
Студент "0..*" - "1..*" Направление
(Студент , Направление) .. Регистрация
@enduml
```

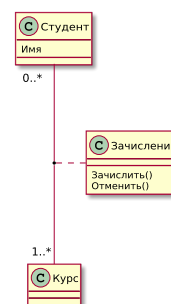
```
Регистрация : принятие()
Регистрация : отмена()
@enduml
```



Можно объявить ассоциацию и в другом направлении:

```
@startuml
Студент : Имя
Студент "0..*" - "1..*" Направление
(Студент , Направление) . Регистрация
@enduml
```

```
Регистрация : принятие()
Регистрация : отмена()
@enduml
```



3.17 Команда «skinparam»

Вы можете использовать команду «skinparam», чтобы изменить цвета и шрифты для диаграммы. Также возможно использование этой команды для:

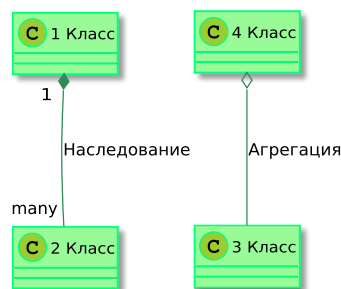
- Определения диаграммы, как и для других любых команд,
- Импортированных файлов,
- Файла конфигурации, вызванного через командную строку.

```
@startuml
skinparam classBackgroundColor PaleGreen
skinparam classArrowColor SeaGreen
skinparam classBorderColor SpringGreen
skinparam stereotypeCBackgroundColor YellowGreen

Класс01 "1" *-- "многие" Класс02 : Наследование

Класс03 o-- Класс04 : Агрегация

@enduml
```



3.18 Разделение больших диаграмм

Иногда, когда получаются очень большие диаграммы, вы можете использовать команду «page (hpages)x(vpages)» для разделения полученного изображения на несколько файлов.

- *hpages* указывает число горизонтальных листов
- *vpages* указывает число вертикальных листов

```
@startuml
' Split into 4 pages
page 2x2

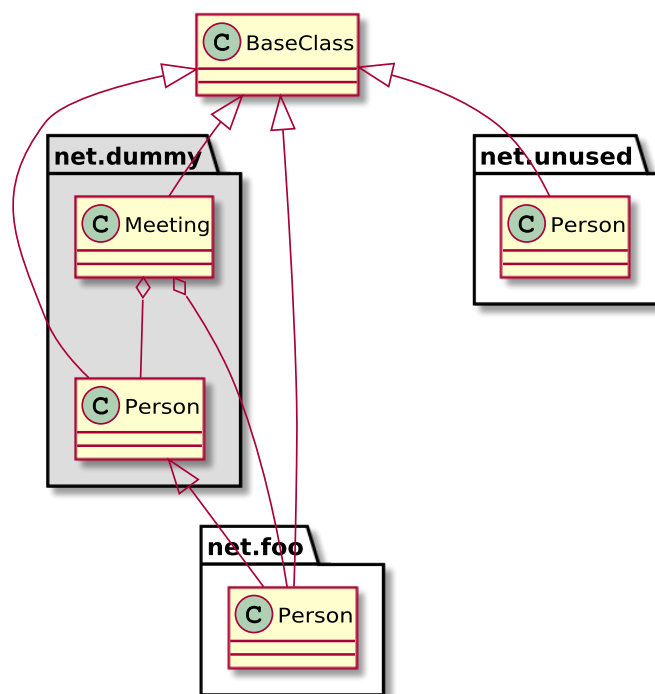
class BaseClass

namespace net.dummy #DDDDDD
  .BaseClass <|-- Person
  Meeting o-- Person

.BaseClass <|-- Meeting
end namespace

namespace net.foo {
  net.dummy.Person <|-- Person
  .BaseClass <|-- Person
}

net.dummy.Meeting o-- Person
}
BaseClass <|-- net.unused.Person
@enduml
```



4. Диаграмма деятельности

4.1 Простая деятельность

Чтобы обозначить начало и конец диаграммы деятельности, используйте символ «*», а для рисования стрелок – «- ->». Пример:

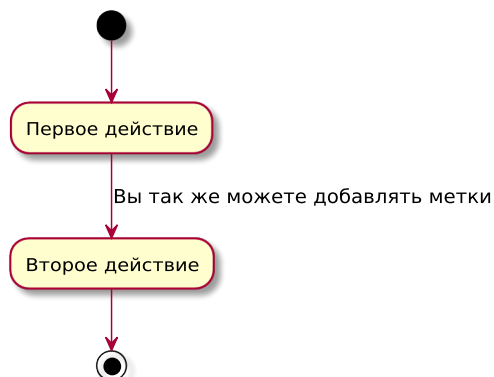
```
@startuml
(*) --> "Первое действие"
"Первое действие" --> (*)
@enduml
```



4.2 Комментарии к стрелкам

По умолчанию, стрелка начинается от последнего действия. Вы можете написать комментарий к стрелочке между квадратными скобками «[» и «]» сразу после описания стрелки в коде.

```
@startuml
(*) --> "Первое действие"
-->[Вы так же можете добавлять метки]
"Второе действие"
--> (*)
@enduml
```

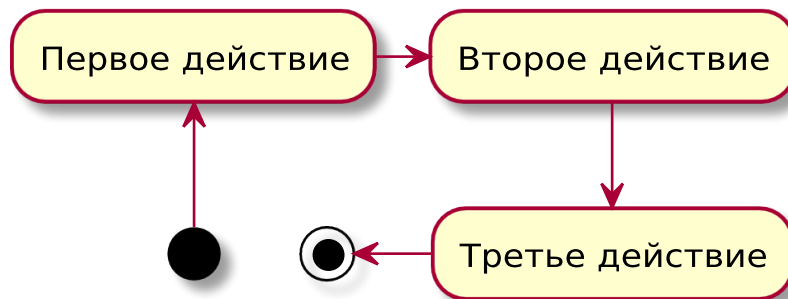


4.3 Изменение направления стрелок

Вы можете использовать «- ->» для построения горизонтальной стрелки. Для изменения ее направления используйте следующий синтаксис:

- -down-> (стрелка по умолчанию)
- -right-> или ->
- -left->
- -up->

```
@startuml
(*) -up-> "Первое действие"
-right-> "Второе действие"
--> "Третье действие"
-left-> (*)
@enduml
```



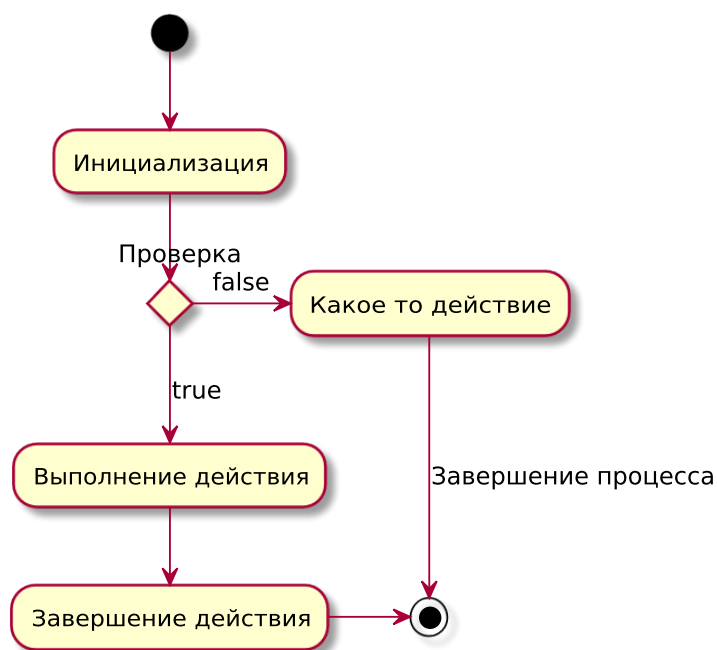
Вы можете сделать стрелки короче, используя в коде только первую букву команды, например, «-d-» вместо «-down-» или первые две буквы «-do-». Обратите внимание на то, что вы не должны перегружать функциональную часть: *Graph Viz* не требует доработок и дает прекрасный результат.

4.4 Разветвление

Используйте команды «if»/«then»/«else» чтобы создать разветвление.

```
@startuml
(*) --> "Инициализация"

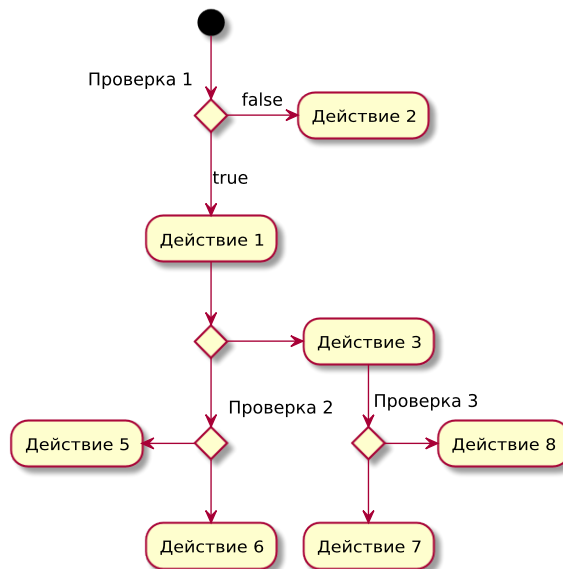
if "Проверка" then
  -->[true] "Выполнение действия"
  --> "Завершение действия"
  -right-> (*)
else
  -->[false] "Какое то действие"
  -->[Завершение процесса] (*)
endif
@enduml
```



4.5 Дополнительно про разветвление

По умолчанию, ветвь соединяется с последним активным объектом, но возможно переделать так, чтобы посредством ключевого слова «if» построить ветвь от другого объекта. Вдобавок, вы можете объединять ветви в группу.

```
@startuml
(*) --> if "Проверка 1" then
  -->[true] "Действие 1"
  if "" then
    -> "Действие 3" as Д3
  else
    if "Проверка 2" then
      -left-> "Действие 5"
    else
      --> "Действие 6"
    endif
  endif
else
  -->[false] "Действие 2"
endif
Д3 --> if "Проверка 3" then
  --> "Действие 7"
else
  -> "Действие 8"
endif
@enduml
```



4.6 Синхронизация

Для того, чтобы нарисовать разделительные полосы для отражения синхронных действий, используйте «=== код ===».

```
@startuml
```

```
(*) --> ===B1===
```

```
--> "Параллельное действие 1"
```

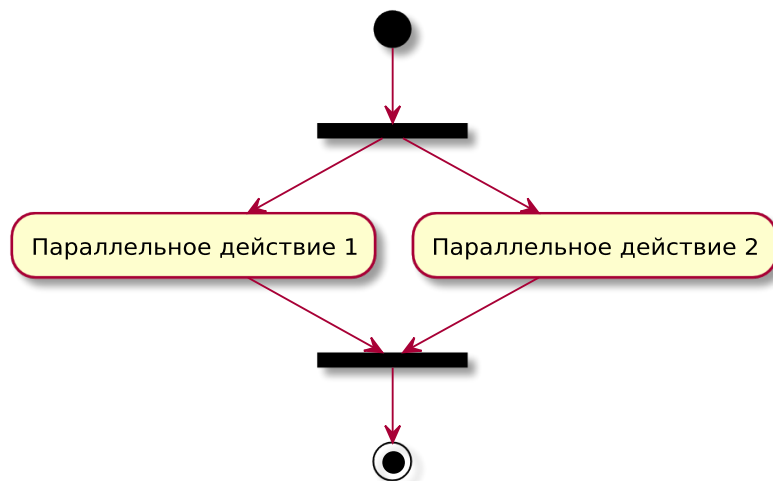
```
--> ===B2===
```

```
===B1=== --> "Параллельное действие 2"
```

```
--> ===B2===
```

```
--> (*)
```

```
@enduml
```



4.7 Полное описание деятельности

При написании схемы диаграммы деятельности, можно использовать многострочные комментарии, при помощи специальных символов «\n». Существует поддержка некоторых html тэгов, таких как:

-
- <i>
- или
- <color:#AAAAAA> или <color:colorName>
- <size:nn> для изменения размера шрифта
- или <img:file> : файл должен быть доступен в файловой системе

Для каждого действия на диаграмме деятельности можно задать псевдоним при помощи команды «as». В дальнейшем можно обращаться по псевдониму, как и показано на примере.

```
@startuml
(*) -left-> "Это <size:20>очень</size>
<b>объемное</b> <color:red>действие,</color>
поэтому может быть описано в несколько
<i>строк</i>" as Д1
-up-> "Другое действие,\n описанное в двух строках"
Д1 --> "Короткое действие <img:F:/1.gif>"
@enduml
```



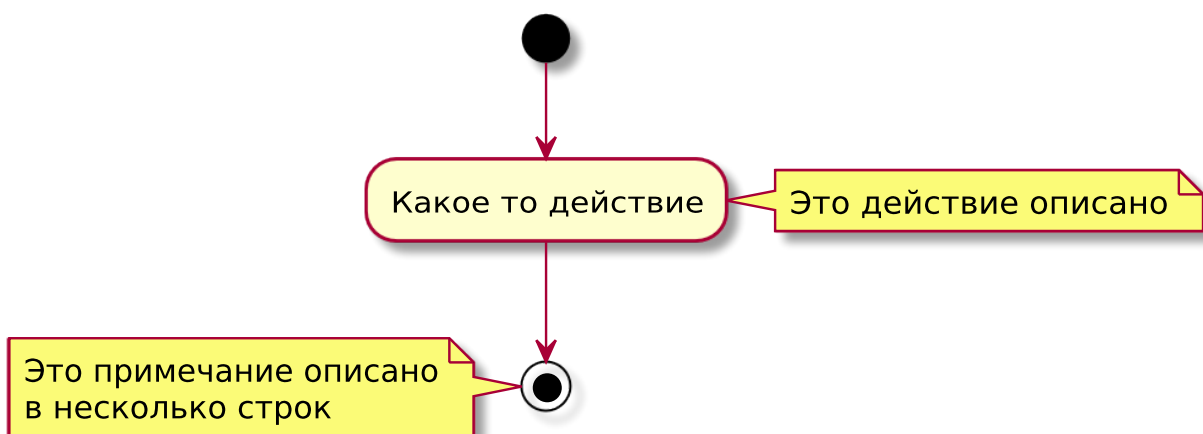
4.8 Заметки

Вы можете добавить заметки к действиям, используя такие команды как:

- note left,
- note right,
- note top,
- note bottom,

после описания действия, к которому вы хотите добавить заметку. Если вы хотите поместить заметку в самом начале диаграммы, то добавьте заметку перед описанием диаграммы. Чтобы разместить многострочную заметку, используйте команду «end note».

```
@startuml
(*) --> "Какое то действие"
note right: Это действие описано
"Какое то действие" --> (*)
note left
  Это примечание описано
  в несколько строк
end note
@enduml
```



4.9 Разделение

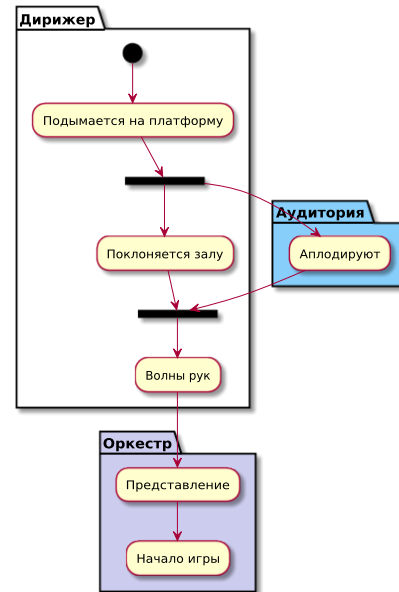
Вы можете разделить диаграмму, используя ключевое слово «**partition**», назначить цвет данной части с помощью HTML кода или просто названия цвета. После объявления действия, оно автоматически попадает в последнюю активную часть. Вы можете завершить разделение, используйте команду «**end partition**».

```
@startuml
partition Дирижер
(*) --> "Поднимается на платформу"
--> === S1 ===
--> "Поклоняется залу"
end partition

partition Аудитория #LightSkyBlue
=== S1 === --> Applauding
end partition

partition Дирижер
"Поклоняется залу"--> === S2 ===
--> "Волны рук"
Аплодируют --> === S2 ===
end partition

partition Оркестр #CCCCEE
"Волны рук" --> "Представление"
--> "Начало игры"
end partition
@enduml
```

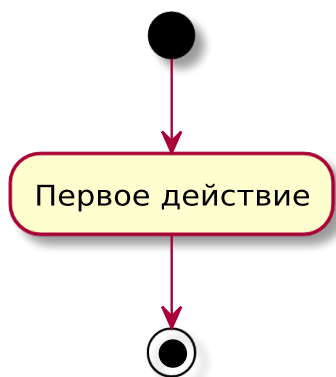


4.10 Название диаграммы

Чтобы дать имя диаграмме используется ключевое слово «`title`». Для длинного названия диаграммы используется сразу две команды «`title`» и «`end title`», как и в случае с диаграммой последовательности.

```
@startuml
title Простой пример \n заголовка
(*) --> "Первое действие"
--> (*)
@enduml
```

Простой пример
заголовка



4.11 Команда «skinparam»

Вы можете использовать команду «**skinparam**», чтобы изменить цвета и шрифты для диаграммы. Также возможно использование этой команды для:

- Определения диаграммы, как и для других любых команд,
- Импортированных файлов,
- Файла конфигурации, вызванного через командную строку.

```
@startuml
skinparam backgroundColor #AFFFFF
skinparam activityStartColor red
skinparam activityBarColor SaddleBrown
skinparam activityEndColor Silver
skinparam activityBackgroundColor Peru
skinparam activityBorderColor Peru
skinparam activityFontName Impact
```

```
(*) --> "Поднимается на платформу"
--> === S1 ===
--> "Поклоняется залу"
--> === S2 ===
--> "Волны рук"
--> (*)
@enduml
```



4.12 Итоговый пример

```

@startuml
'http://click.sourceforge.net/images/activity-diagram-small.png
title Контейнер сервлетов

(*) --> "ClickServlet.handleRequest()"
--> "new Page"

if "SecurityCheck" then
->[true] "Page.onInit()"
if "isForward?" then
->[нет] "Процесс контроля"
if "продолжить?" then
-->[да] ===RENDERING===
else
-->[нет] ===REDIRECT_CHECK===
endif
else
-->[да] ===RENDERING===
endif
if "is Post?" then
-->[да] "Page.onPost()"
--> "Page.onRender()" as render
--> ===REDIRECT_CHECK===
else
-->[нет] "Page.onGet()"
--> render
endif
else
-->[false] ===REDIRECT_CHECK===
endif
if "Перенаправление?" then
->[да] "Перенаправление запроса"
--> ==BEFORE_DESTROY==
else
if "Отправка запроса?" then
-left->[да] "Отправка запроса"
--> ==BEFORE_DESTROY==
else
-right->[нет] "Рендеринг шаблона страницы"
--> ==BEFORE_DESTROY==
endif
endif
--> "Page.onDestroy()"
-->(*)
@enduml

```

