

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

PlantUML > Language specification > Activity Diagram (beta) >

sponsors 20

patreon 122

liberapay 10

paypal 296

Диаграмма активности

Старый синтаксис диаграммы активности имеет несколько ограничений и недостатков (например, её сложно поддерживать).

На замену этому старому синтаксису здесь представлен новый синтаксис. Хотя старый синтаксис будет по-прежнему поддерживаться, чтобы обеспечивать обратную совместимость, для новых проектов пользователям рекомендуется переходить на новый синтаксис.

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

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

Простая активность

Описания активностей начинаются с `:` и заканчиваются `;`. Само описание может содержать в себе несколько строчек.

Как уже упоминалось выше, дополнительно форматировать текст описания активности возможно, используя синтаксис `creole`.

Активности автоматически связываются по порядку их определения.

@startuml

:Привет мир!;

:Эта активность описывается несколькими **строчками** текста;

:|= Еще один | = быстрый пример |

| использования | creole разметки |;

@enduml

PlantUML 1.2023.2beta1

[From string (line 9)]

@startuml

:Привет мир!;

:Эта активность описывается несколькими **строчками** текста;

:|= Еще один | = быстрый пример |

| использования | creole разметки |;

Syntax Error?

- Простая активность
- Старт активности, Стоп активности, Стоп поток, Быстрый стоп
- Условия (оператор if)
- Перебор множества условий для определенного параметра (оператор switch)
- Остановка активности в теле условия
- Повторяющийся цикл (с проверкой условия в конце цикла)
- Прерывание цикла repeat при помощи ключевого слова break
- Goto and Label Processing [label, goto]
- Повторяющийся цикл (с проверкой условия в начале цикла)
- Распараллеливание активности [режим многопоточности: fork, fork again, end fork и merge]
- Распараллеливание активности [режим разделения: split, split again и end split]
- Заметки
- Цвета
- Линии вместо стрелок
- Изменение цвета и стиля стрелок (синтаксис квадратных скобочек)
- Коннектор
- Цвет коннектора
- Объединение (группировка) активностей
- Дорожки
- Прерывание активности
- SDL (язык спецификаций)

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

Старт активности, Стоп активности, Стоп поток, Быстрый стоп

Следующие ключевые слова используются для:

- start - для обозначения начального состояния активности,
- stop - для обозначения конечного состояния активности,
- end - для обозначения конечного состояния потока,
- kill или detach - для быстрого прерывания активности без обозначения каких либо значков.

Использование любого из них не обязательно, диаграммы активности можно строить и без указания начального и/или конечного состояний start , stop или end , как было продемонстрировано на примерах выше. Чтобы прервать какую-либо активность без отображения каких-либо значков, нужно использовать идентичные по своему действию ключевые слова kill или detach .

```
@startuml
start
:Привет мир!;
:Эта активность описывается
несколькими **строчками** текста;
stop
@enduml
```



```
@startuml
start
:Привет мир!;
:Эта активность описывается
несколькими **строчками** текста;
end
@enduml
```



```
@startuml
if (условие) then
:Один\ntkill;
kill
else
:Два;
endif
:Три\nstop;
stop
@enduml
```



Условия (оператор if)

Можно использовать ключевые слова `if`, `then`, `else` и `endif`, чтобы добавить проверку условия на диаграмму.

Доступно 3 равнозначных варианта записи условия (квадратными скобками отмечены опциональные поля, которые можно указывать, а можно не указывать):

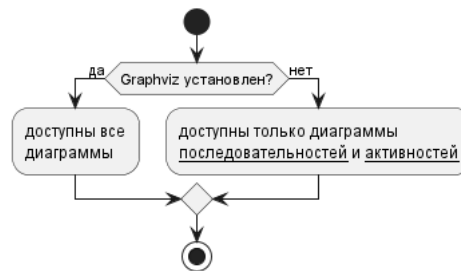
1 вариант записи:

- `if (текстУсловия) then [(подписьУспешно)]`
- `else [(подписьНеУспешно)]`
- `endif`

```

@startuml
start
if (Graphviz установлен?) then (да)
    :доступны все\н\диаграммы;
else (нет)
    :доступны только диаграммы
    __последовательностей__ и __активностей__;
endif
stop
@enduml

```



2 вариант записи

- `if (текстУсловия) is (подписьУспешно) then`
- `else [(подписьНеУспешно)]`
- `endif`

```

@startuml
if (цвет?) is (<color:red>красный) then
    :печатаем "красный";
else
    :печатаем "НЕ красный";
endif
@enduml

```



3 вариант записи

- `if (текстУсловия) equals (подписьУспешно) then`
- `else [(подписьНеУспешно)]`
- `endif`



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

```
if (счетчик?) is (5) then
  :пачатаем "5";
else
  :печатаем "НЕ 5";
endif
@enduml
```



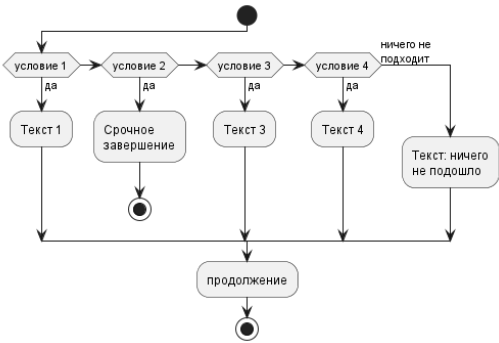
[Ref. [QA-301](#)]

Ряд последовательных проверок условий (горизонтальное отображение)

Вы можете использовать ключевое слово `elseif` , чтобы создать несколько последовательных проверок условий (по умолчанию используется режим горизонтального отображения):

```
@startuml
start
if (условие 1) then (да)
  :Текст 1;
elseif (условие 2) then (да)
  :Срочное\пзавершение;
  stop
elseif (условие 3) then (да)
  :Текст 3;
elseif (условие 4) then (да)
  :Текст 4;
else (ничего не\пподходит)
  :Текст: ничего\пне подошло;
endif

:продолжение;
stop
@enduml
```



Ряд последовательных проверок условий (вертикальное отображение)

Используйте опцию `!pragma useVerticalIf on` для отображения нескольких последовательных проверок условий в вертикальном режиме:

-
-

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

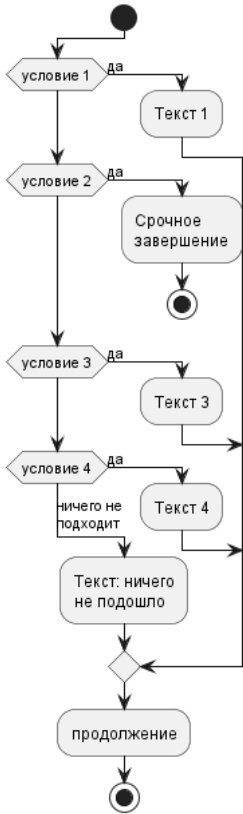
```
!pragma useVerticalIf on
start

if      (условие 1) then (да)
  :Текст 1;
elseif (условие 2) then (да)
  :Срочное\завершение;
  stop
elseif (условие 3) then (да)
  :Текст 3;
elseif (условие 4) then (да)
  :Текст 4;
else (ничего не\подходит)
  :Текст: ничего\пне подошло;
endif

:продолжение;

stop

@enduml
```



Также, можно использовать параметр -P для установки этой опции для диаграмм по умолчанию из [командной строки](#):

```
java -jar plantuml.jar -PuseVerticalIf=on
```

[Refs. [QA-3931](#), [issue-582](#)]

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

Перебор множества условий для определенного параметра (оператор switch)

Для добавления в диаграмму перебора множества условий для какого-то одного параметра (конструкция switch-case), нужно использовать следующие ключевые слова: switch , case , endswitch .

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

```
switch ( параметрДляКоторогоПроверяютсяУсловияНиже )
case ( условие1 ? )
:вариант 1;
case ( условие2 ? )
:вариант 2;
...
endswitch
```

⌵

@startuml

start

switch (число потоков)

case (равно 1)

:1 поток;

case (равно 2)

:2 потока;

case (равно 3)

:3 потока;

case (равно 4)

:4 потока

срочное

завершение

kill;

kill

case (равно 5)

:5 потоков;

endswitch

stop

@enduml

Остановка активности в теле условия

Как уже было продемонстрировано ранее, можно остановить активность на конкретном действии, используя ключевые слова stop для отображения остановки деятельности или end для отображения остановки потока.

⌵

@startuml

if (условие?) then

:ошибка;

stop

endif

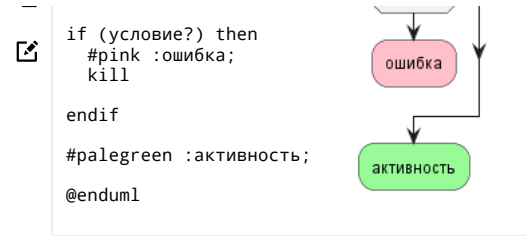
#palegreen :активность;

@enduml

Если нужно прервать активность на каком-то конкретном действии без отображения каких-либо значков, нужно использовать по своему действию ключевые слова kill или detach :

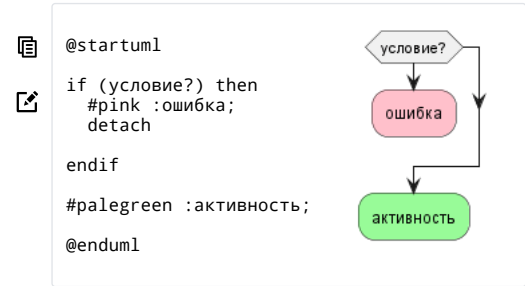
- kill

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



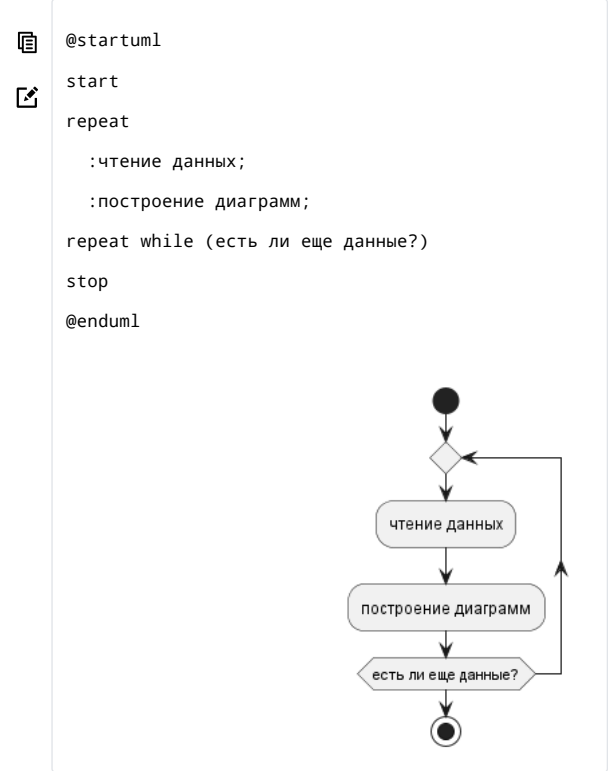
[Ref. [QA-265](#)]

- detach



Повторяющийся цикл (с проверкой условия в конце цикла)

Для отображения циклов нужно использовать идентичные по своему действию ключевые слова `repeat` или `repeatwhile` :



Чтобы добавить подписи к стрелочкам после проверки условия, нужно использовать следующую конструкцию:

- `repeat while (условие) is (подписьКогдаУспешно)`
`not (подписьКогдаНеУспешно)`

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

```
start
repeat
    :чтение данных;
    :построение диаграмм;
repeat while (есть ли еще данные?) is (да) not (нет)
stop
@enduml
```



Можно превратить начало цикла в полноценное действие, если после ключевого слова `repeat` в той же строке указать это действие при помощи конструкции `:действие`;

```
@startuml
start
repeat :начало цикла;
    :чтение данных;
    :построение диаграмм;
repeat while (есть ли еще данные?)
stop
@enduml
```



Чтобы добавить к циклу обратное действие, используйте ключевое слово `backward` и сразу в той же строке укажите это действие при помощи конструкции `:действие`;

-
-

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

```
start
repeat :начало цикла;
      :чтение данных;
      :построение диаграмм;
backward :очистка экрана;
repeat while (есть ли еще данные?)

stop
@enduml
```



⌵ [🔗](#) **Прерывание цикла repeat при помощи ключевого слова break**

Чтобы прервать цикл после определенного действия, нужно использовать ключевое слово `break`



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

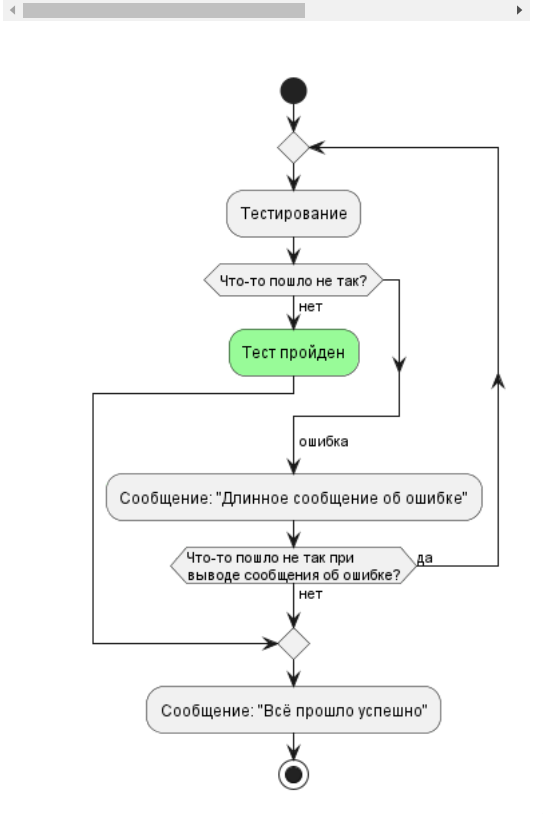
```
start
repeat
  :Тестирование;

  if (Что-то пошло не так?) then (нет)
    #palegreen :Тест пройден;
    break
  endif

  ->ошибка;

  :Сообщение: "Длинное сообщение об ошибке";

repeat while (Что-то пошло не так при\выводе соо
:Сообщение: "Всё прошло успешно";
stop
@enduml
```



[Ref. [QA-6105](#)]

Goto and Label Processing [label, goto]

⚠ It is currently only experimental 🚧

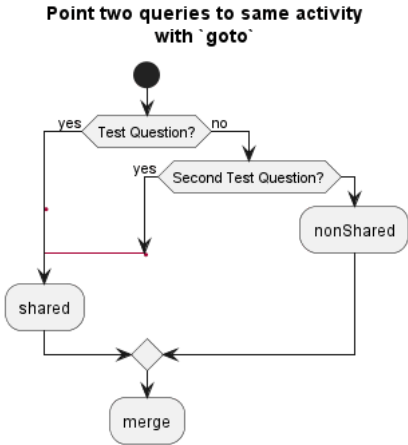
You can use `label` and `goto` keywords to denote goto processing, with:

- `label <label_name>`
- `goto <label_name>`



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

```
title Point two queries to same activity\nwith `g`\nstart\nif (Test Question?) then (yes)\n'space label only for alignment\nlabel sp_lab0\nlabel sp_lab1\n'real label\nlabel lab\n:shared;\nelse (no)\nif (Second Test Question?) then (yes)\nlabel sp_lab2\ngoto sp_lab1\nelse\n:nonShared;\nendif\nendif\n:merge;\n@enduml
```

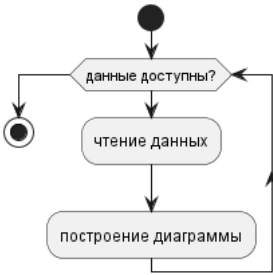


[Ref. [QA-15026](#), [QA-12526](#) and initially [QA-1626](#)]

Повторяющийся цикл (с проверкой условия в начале цикла)

Для организации цикла с проверкой условия в начале используйте ключевые слова `while` и `end while`.

```
@startuml\nstart\nwhile (данные доступны?)\n:чтение данных;\n:построение диаграммы;\nendwhile\nstop\n@enduml
```



Чтобы добавить пояснения на стрелочки после проверки условия, нужно использовать следующую конструкцию:

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

...
endwhile (пояснениеКогдаНеУспешно)

Пример:

```
@startuml
while (данные доступны?) is (да)
    :чтение данных;
    :построение диаграммы;
endwhile (нет)
:окончание работы;
@enduml
```

Чтобы добавить к циклу обратное действие, используйте ключевое слово `backward` и сразу в той же строке укажите это действие при помощи конструкции `:действие ;`

```
@startuml
while (данные доступны?) is (да)
    :чтение данных;
    :построение диаграммы;
    backward :очистка экрана;
endwhile (нет)
:окончание работы;
@enduml
```

Чтобы прервать выполнение цикла, используйте ключевое слово `break` :

```
@startuml
while (данные доступны?) is (да)
    :чтение данных;
    :построение диаграммы;
    break;
endwhile (нет)
:окончание работы;
@enduml
```

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

```
while (данные доступны?) is (да)
  :чтение данных;
  if (данные прочитаны?) then (нет)
    #pink :Сообщение об ошибке;
    break
  endif
  :построение диаграммы;
  backward :очистка экрана;
endwhile (нет)
->завершение;
#palegreen :окончание работы;
@enduml
```



WARNING This translation need to be updated. WARNING

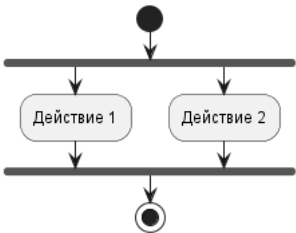
Распаралелливание активности [режим многопоточности: fork, fork again, end fork и merge]

Для разделения одной активности на несколько параллельных активностей можно использовать режим многопоточности. Для этого необходимо использовать ключевые слова: fork , fork again и end fork или end merge .

(Также для этих целей можно использовать режим разделения (split), который будет рассмотрен более подробно ниже).

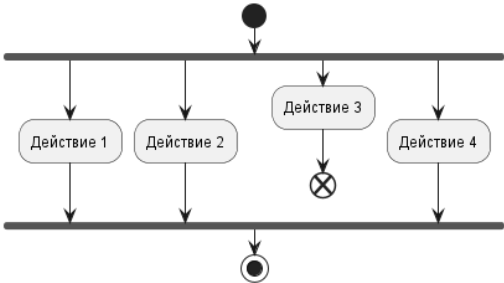
Простой режим многопоточности

```
@startuml
start
fork
  :Действие 1;
fork again
  :Действие 2;
end fork
stop
@enduml
```



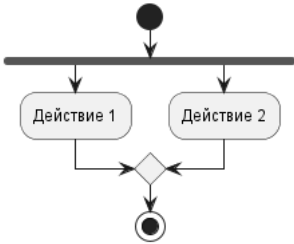
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
start
fork
:Действие 1;
fork again
:Действие 2;
fork again
:Действие 3;
end
fork again
:Действие 4;
end fork
stop
@enduml
```

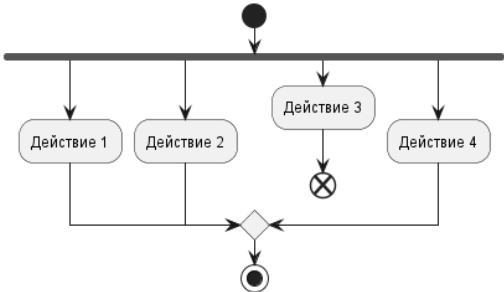


Режим многопоточности с объединением потоков в конце

```
@startuml
start
fork
:Действие 1;
fork again
:Действие 2;
end merge
stop
@enduml
```



```
@startuml
start
fork
:Действие 1;
fork again
:Действие 2;
fork again
:Действие 3;
end
fork again
:Действие 4;
end merge
stop
@enduml
```



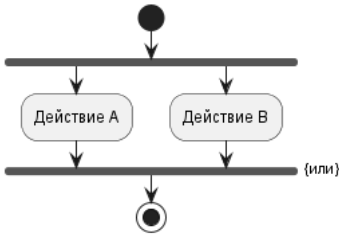
[Ref. [QA-13731](#)]

Размещение комментария при возвращении к однопоточному режиму (реализация синтаксиса UML):

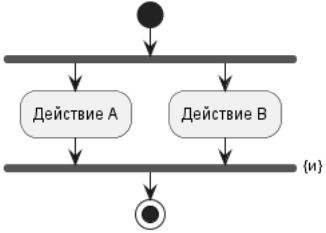
-
-

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

```
start
fork
:Действие A;
fork again
:Действие B;
end fork {или}
stop
@enduml
```



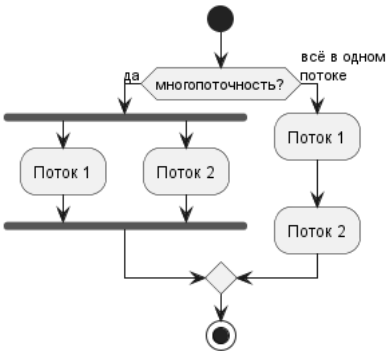
```
@startuml
start
fork
:Действие A;
fork again
:Действие B;
end fork {и}
stop
@enduml
```



[Ref: QA-5346]

Еще один пример

```
@startuml
start
if (многопоточность?) then (да)
fork
:Поток 1;
fork again
:Поток 2;
end fork
else (всё в одном\потоке)
:Поток 1;
:Поток 2;
endif
stop
@enduml
```

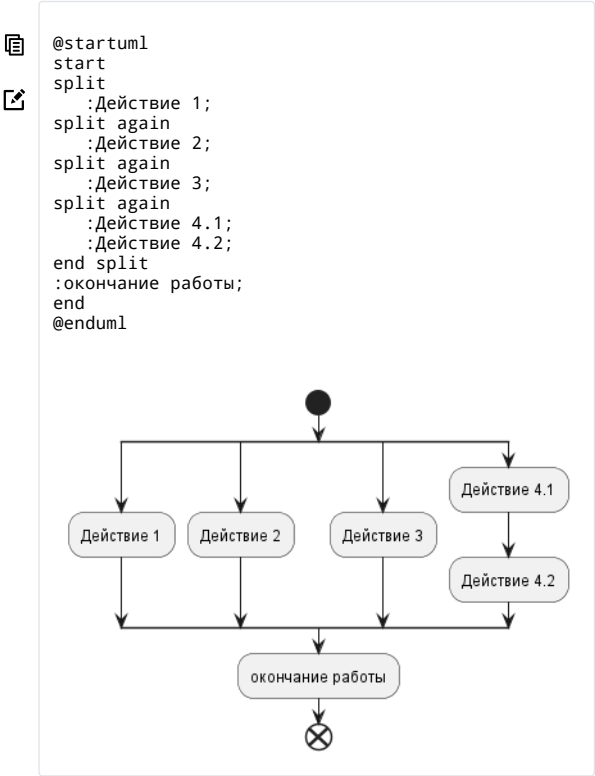


Распараллеливание активности [режим разделения: split, split again и end split]

Разделение

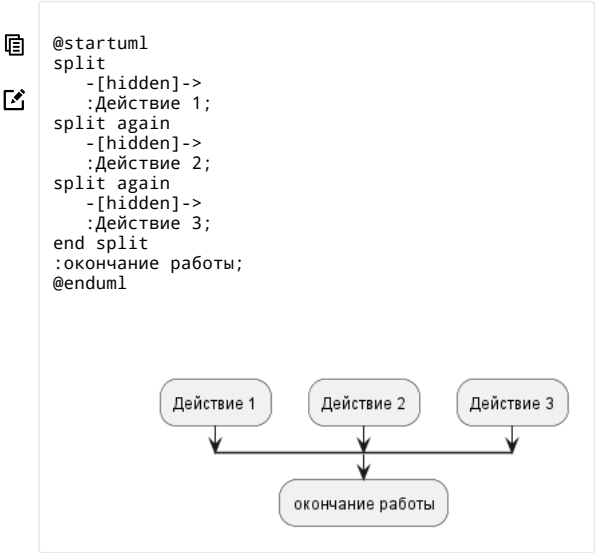
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

разделения. Для этого необходимо использовать следующие ключевые слова: split , split again и end split .



Несколько входов

При помощи режима разделения можно изобразить несколько независимых начальных действий. Для этого необходимо организовать распараллеливание активности на несколько параллельных и при этом скрыть входные стрелочки при помощи добавления конструкции -[hidden]-> перед каждой активностью:

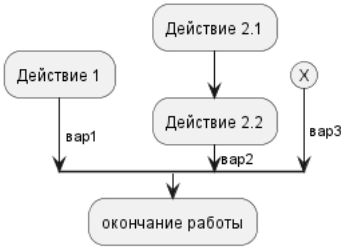


Более сложный пример:



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

```
split
-[hidden]->
:Действие 1;
->вар1;
split again
-[hidden]->
:Действие 2.1;
:Действие 2.2;
->вар2;
split again
-[hidden]->
(X)
->вар3;
end split
:окончание работы;
@enduml
```



[Ref. [QA-8662](#)]

Несколько выходов

Таким же образом можно организовать несколько параллельно завершаемых действий. Для этого после каждого такого действия нужно указать ключевые слова `kill` или `detach` :

```
@startuml
start
split
:Действие 1;
kill
split again
:Действие 2;
detach
split again
:Действие 3;
kill
end split
@enduml
```



Более сложный пример:

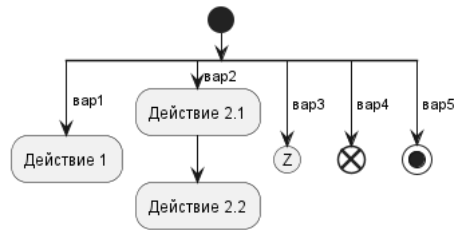
-
-

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

```

start
split
->var1;
:Действие 1;
kill
split again
->var2;
:Действие 2.1;
:Действие 2.2;
detach
split again
->var3;
(Z)
detach
split again
->var4;
end
split again
->var5;
stop
end split
@enduml

```



Заметки

На диаграмму активностей (также, как и на остальные диаграммы) можно добавлять заметки.

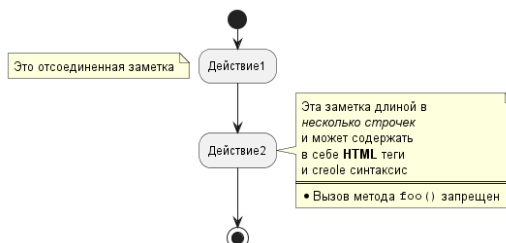
Текст внутри заметок можно дополнительно форматировать используя [creole](#) и [HTML](#) синтаксис.

По умолчанию создаются заметки, которые явно указывают на элемент, который был объявлен непосредственно перед объявлением заметки. Чтобы создать отсоединенную заметку, нужно использовать ключевое слово `floating` (она по прежнему будет находится рядом с элементом, но у неё не будет "хвостика" указателя).

```

@startuml
start
:Действие1;
floating note left: Это отсоединенная заметка
:Действие2;
note right
  Эта заметка длиной в
  //несколько строчек//
  и может содержать
  в себе <b>HTML</b> теги
  и creole синтаксис
  ====
  * Вызов метода ""foo()"" запрещен
end note
stop
@enduml

```



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

```
@startuml
start
  repeat :Ввод данных;
  :Отправка;
  backward :Предупреждение;
  note right: Заметка о том\nчто что-то пошло
  repeat while (Корректно?) is (Нет) not (Да)
  stop
@enduml
```

[Ref. [QA-11788](#)]

Можно добавлять заметки к разделам:

```
@startuml
start
partition "***процесс** Привет Мир" {
  note
    Тут наша заметка
    ---
    //Creole тест//
  end note
  :Готовность к отправке;
  : "Привет Мир" -> получатель X>
  :Привет отправлен;
}
@enduml
```

[Ref. [QA-2398](#)]

Цвета

Можно задавать цвета для активностей

-
-

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

```
:запуск процесса;  
  
#HotPink:чтение конфигурационных файлов  
----  
На этом этапе указанные файлы  
уже должны быть отредактированы!;  
  
#AAAAAA:остановка процесса;  
  
@endum1
```

Также можно задавать заливку [градиентом](#):

```
@startuml  
start  
  
partition #red/white "тестирование раздела" {  
  #lightblue\palegreen :тестирование активности;  
}  
  
stop  
@endum1
```

[Ref: [QA-4906](#)]

Линии вместо стрелок

Можно превратить все стрелки на диаграмме в линии, если указать команду `skinparam ArrowHeadColor none` в начале диаграммы.

-
-

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

```
skinparam ArrowHeadColor none
start
:Привет Мир!;
:Какое-то действие в
несколько **строчек**;;
stop
@enduml
```



```
@startuml
skinparam ArrowHeadColor none
start
repeat :Ввод данных;
:Отправка;
backward :Предупреждение;
repeat while (Коректно?) is (Нет) not (Да)
stop
@enduml
```



✖ Изменение цвета и стиля стрелок (синтаксис квадратных скобочек)

Явно объявляя стрелку и указывая для неё дополнительные параметры, можно управлять отображением стрелок между элементами диаграммы.

Стиль линий

Можно задавать различные стили отображения линий при помощи следующей конструкции:

- - [#типЛинии] ->

где типЛинии может принимать значения:

- bold - жирный,
- dashed - штриховой,
- dotted - пунктирный,
- hidden - скрытый,
- plain - непрерывный.

Заметки на линиях

Можно добавлять текстовые подписи к линиям, используя следующую конструкцию:

- --> Текст заметки

Цвет линий

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

• -[#цветЛинии]->

где цветЛинии может принимать текстовые **обозначения цветов** или их числовое представление.

Полный синтаксис

Можно смешивать все указанные выше параметры, следующим образом:

• -[#цветЛинии,типЛинии]-> Текст заметки

Комплексный пример

```
@startuml
:Действие1;
-> Можно добавлять\ntекст на стрелки;
if (Условие?) then
-#blue->
:Действие2;
-#green,dashed-> Текст с //пояснениями//
на стрелках может быть
**очень** длинным...;
:Действие3;
else
-#black,dotted->
:Действие4.1;
-[hidden]->
:Действие4.2;
endif
-#gray,bold->
:Действие5;
@enduml
```

Коннектор

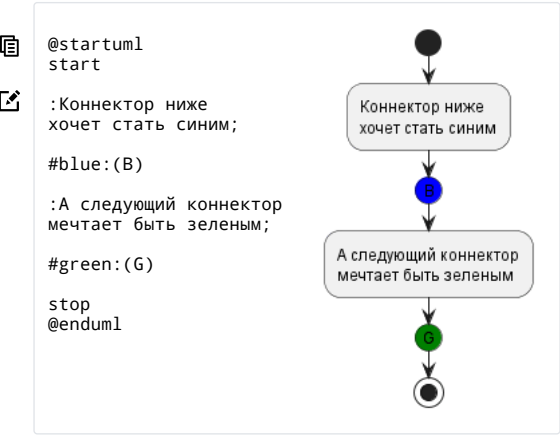
Можно добавлять на диаграмму специальный элемент - коннектор. Для этого необходимо использовать круглые скобки. Чтобы убрать связь между двумя коннекторами, нужно использовать ключевое слово detach .

```
@startuml
start
:Некоторое действие;
(A)
detach
(A)
:Следующее действие;
@enduml
```

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

Цвет коннектора

Можно указать **цвет** для коннектора.



[Ref: [QA-10077](#)]

Объединение (группировка) активностей

Можно объединять (группировать) активности на диаграмме используя:

- group - группы;
- partition - разделы;
- package - пакеты;
- rectangle - прямоугольники;
- card - карточки.

отличаются эти варианты только внешним видом (а для group еще немного другим синтаксисом записи и отсутствием возможности задавать цвет).

Группы

Итак, можно объединять активности используя группы:

```
group НазваниеГруппы
.. объединяемые активности ..
end group
```

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

```
start
group Инициализация
:чтение конфигурационного файла;
:инициализация внутренних переменных;
end group
group Выполнение
:ожидание активности пользователя;
:вывод информации;
end group
stop
@enduml
```



Разделы, Пакеты, Прямоугольники и Карточки

Также, можно объединять активности используя Разделы, Пакеты, Прямоугольники и Карточки. Синтаксис у всех этих вариантов одинаковый, отличие состоит только в форме отображения.

```
partition/package/rectangle/card [#цвет] Название {
.. объединяемые активности ..
}
```



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

```
start
group Группа
:Действие;
end group
floating note: Заметка для группы

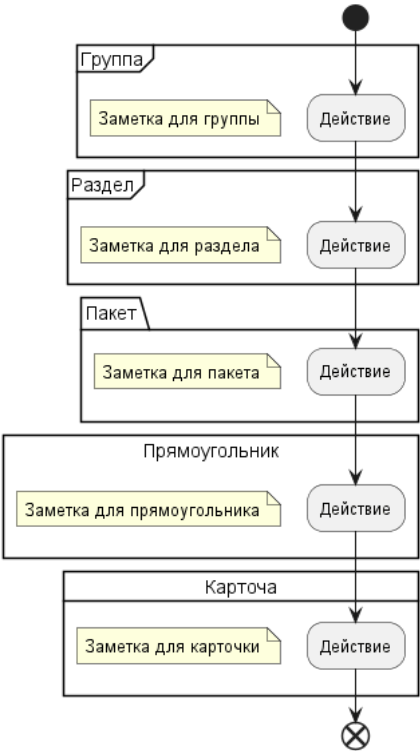
partition Раздел {
:Действие;
}
floating note: Заметка для раздела

package Пакет {
:Действие;
}
floating note: Заметка для пакета

rectangle Прямоугольник {
:Действие;
}
floating note: Заметка для прямоугольника

card Карточка {
:Действие;
}
floating note: Заметка для карточки

end
@enduml
```



Для Разделов, Пакетов, Прямоугольников и Карточек можно задавать [цвет](#) отображения (для Групп этого сделать нельзя):



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

```
start

partition #lightGreen Инициализация {
  :чтение конфигурационного файла;
  :инициализация внутренних переменных;
}

partition Выполнение {
  :ожидание активности пользователя;
  :вывод информации;
}

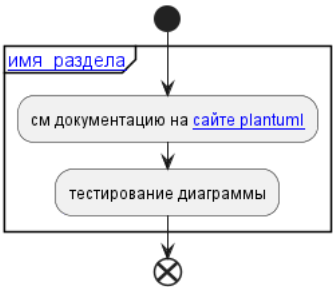
stop
@enduml
```



[Ref. [QA-2793](#)]

Также, в названиях можно использовать [гиперссылки](#):

```
@startuml
start
partition "[[http://plantuml.com имя_раздела]]" {
  :см документацию на [[http://plantuml.com сай
  :тестирование диаграммы;
}
end
@enduml
```



[Ref. [QA-542](#)]

Дорожки

Можно задать дорожки используя символ `|` . Также для задаваемых дорожек можно задавать [цвета](#).



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

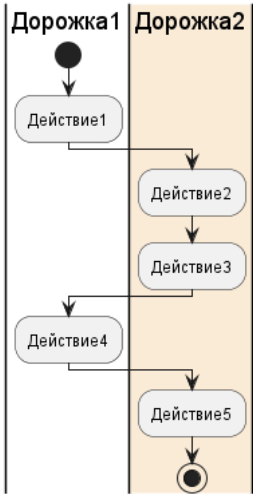
```
|Дорожка1|
start
:Действие1;

|#AntiqueWhite|Дорожка2|
:Действие2;
:Действие3;

|Дорожка1|
:Действие4;

|Дорожка2|
:Действие5;
stop

@endum1
```



Можно использовать дорожки вместе с Условиями и Циклами.

```
@startuml

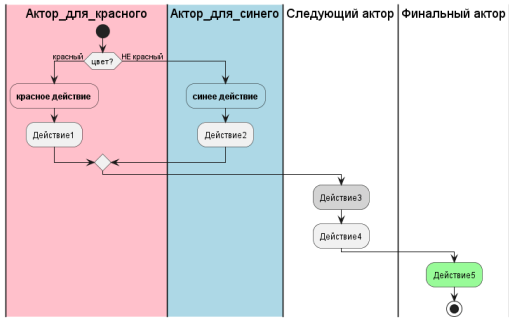
|#pink|Актор_для_красного|
start
if (цвет?) is (красный) then
#pink : **красное действие**
:Действие1;
else (НЕ красный)

|#lightblue|Актор_для_синего|
#lightblue : **синее действие**
:Действие2;
endif

|Следующий актор|
#lightgray :Действие3;
:Действие4;














|Финальный актор|
#palegreen :Действие5;
stop

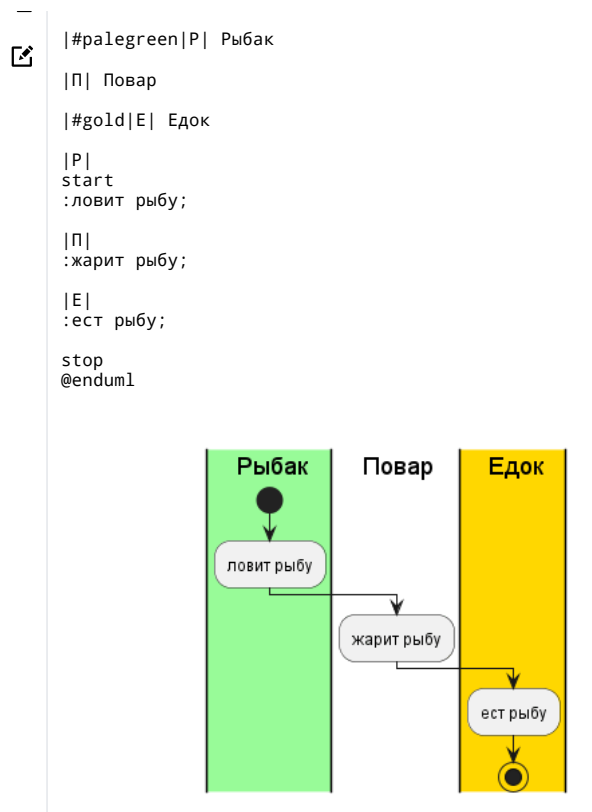
@enduml
```



Также, для того, чтобы каждый раз не писать длинное имя Дорожки, можно задать для дорожки короткое имя и далее использовать везде его. Сделать это можно используя ключевое слово `alias` при первом определении Дорожки:

- `|[#<color>]|<короткоеИмя>|<ДлинноеИмя>`

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



[Ref. [QA-2681](#)]

Прерывание активности

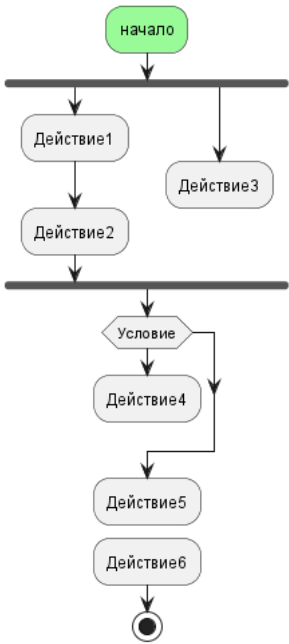
Можно прервать активность (удалить стрелку) используя идентичные по своему действию ключевые слова `detach` или `kill` :

- detach



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

```
#palegreen :начало;  
  
fork  
  :Действие1;  
  :Действие2;  
  
fork again  
  :Действие3;  
  detach  
endfork  
  
if (Условие) then  
  :Действие4;  
  detach  
endif  
  
:Действие5;  
detach  
:Действие6;  
stop  
@endum1
```

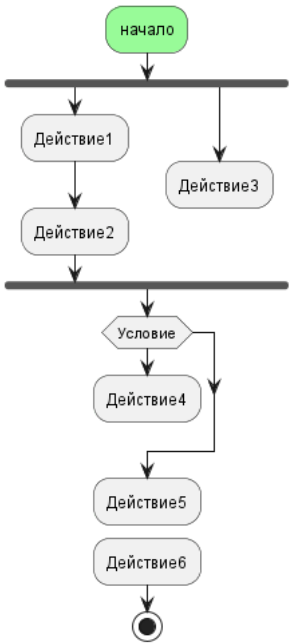


• kill



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

```
#palegreen :начало;  
  
fork  
:Действие1;  
:Действие2;  
  
fork again  
:Действие3;  
kill  
endfork  
  
if (Условие) then  
:Действие4;  
detach  
endif  
  
:Действие5;  
detach  
:Действие6;  
stop  
@endum1
```



SDL (язык спецификаций с формальной семантикой)

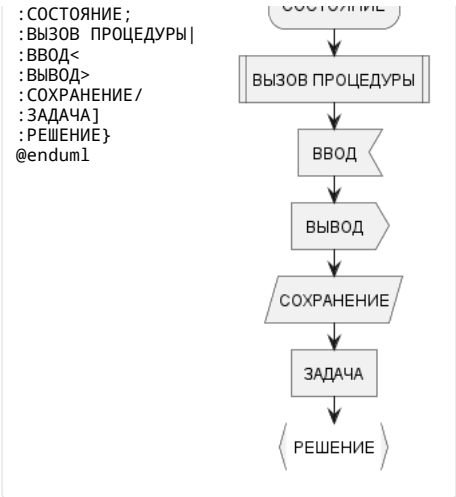
(еще на русский переводится как "язык спецификаций и описаний")

Заменяя символ ; , обозначающий конец описания действия, на символы из списка ниже, можно изменять внешний вид действия. Ниже перечислены доступные варианты с указанием обозначаемых ими типов:

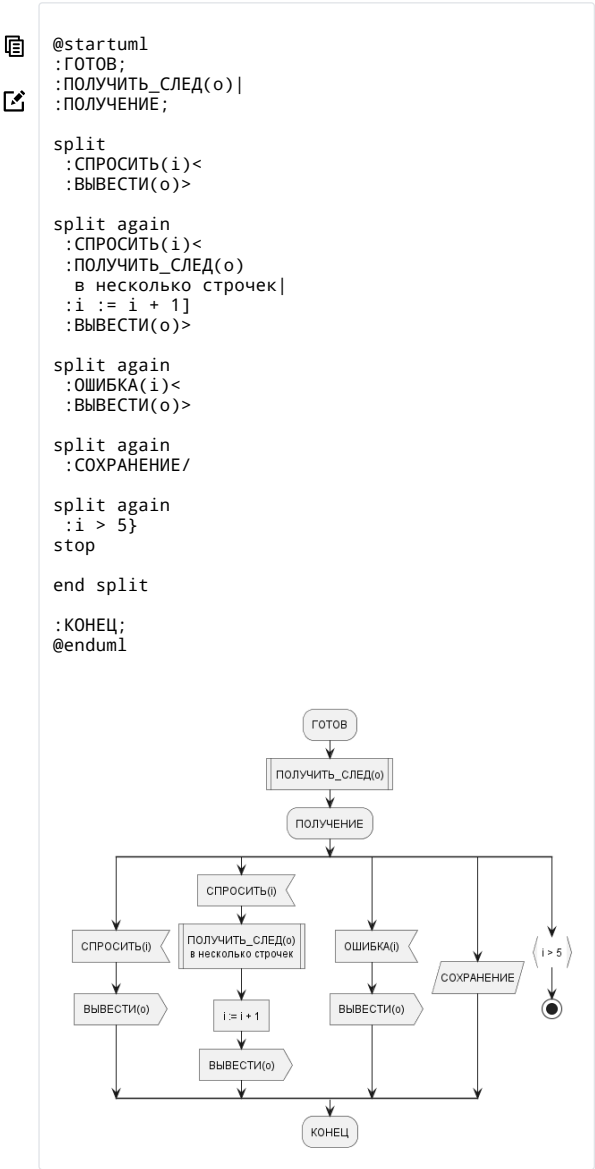
- ; - STATE , СОСТОЯНИЕ
- | - CALL , ВЫЗОВ ПРОЦЕДУРЫ
- < - INPUT , ВВОД
- > - OUTPUT , ВЫВОД
- / - SAVE , СОХРАНЕНИЕ
-] - TASK , ЗАДАЧА
- } - DECISION , РЕШЕНИЕ



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



Еще один пример:



✎ Полноценный пример



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

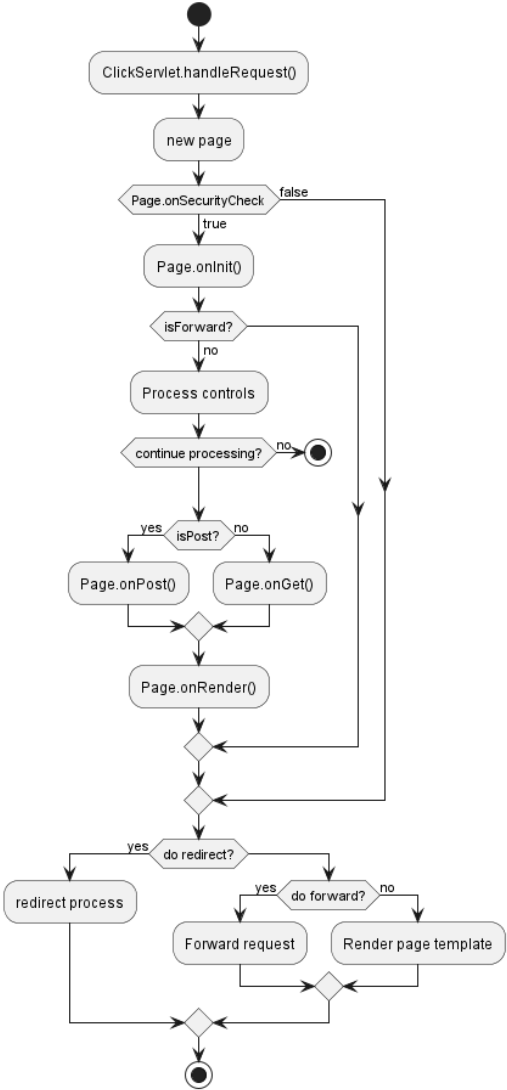
```
start
:ClickServlet.handleRequest();
:new page;
if (Page.onSecurityCheck) then (true)
:Page.onInit();
if (isForward?) then (no)
:Process controls;
if (continue processing?) then (no)
stop
endif

if (isPost?) then (yes)
:Page.onPost();
else (no)
:Page.onGet();
endif
:Page.onRender();
endif
else (false)
endif

if (do redirect?) then (yes)
:redirect process;
else
if (do forward?) then (yes)
:Forward request;
else (no)
:Render page template;
endif
endif
endif

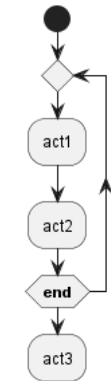
stop

@enduml
```

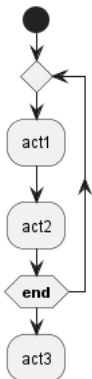


- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
@startuml
skinparam conditionStyle inside
start
repeat
:act1;
:act2;
repeatwhile (<b>end)
:act3;
@enduml
```

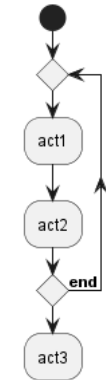


```
@startuml
start
repeat
:act1;
:act2;
repeatwhile (<b>end)
:act3;
@enduml
```



Diamond style

```
@startuml
skinparam conditionStyle diamond
start
repeat
:act1;
:act2;
repeatwhile (<b>end)
:act3;
@enduml
```



InsideDiamond (or Foo1) style



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

```
skinparam conditionStyle InsideDiamond
start
repeat
    :act1;
    :act2;
repeatwhile (<b>end)
:act3;
@enduml1
```

```
@startuml
skinparam conditionStyle foo1
start
repeat
    :act1;
    :act2;
repeatwhile (<b>end)
:act3;
@enduml1
```

[Ref. [QA-1290](#) and [#400](#)]

Condition End Style

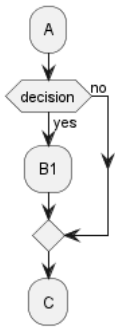
Diamond style (by default)

- With one branch



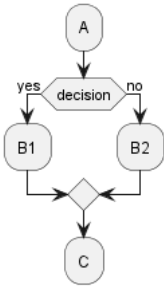
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam ConditionEndStyle diamond
:A;
if (decision) then (yes)
    :B1;
else (no)
    :C;
endif
@enduml
```



- With two branches (B1 , B2)

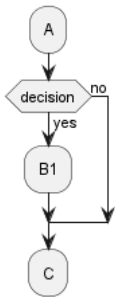
```
@startuml
skinparam ConditionEndStyle diamond
:A;
if (decision) then (yes)
    :B1;
else (no)
    :B2;
endif
:C;
@enduml
```



Horizontal line (hline) style

- With one branch

```
@startuml
skinparam ConditionEndStyle hline
:A;
if (decision) then (yes)
    :B1;
else (no)
    :C;
endif
@enduml
```



- With two branches (B1 , B2)

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

```
skinparam ConditionEndStyle hline
:A;
if (decision) then (yes)
  :B1;
else (no)
  :B2;
endif
:C;
@enduml
@enduml
```

[Ref: [QA-4015](#)]

Using (global) style

Without style (by default)

```
@startuml
start
:init;
-> test of color;
if (color?) is (<color:red>red) then
:print red;
else
:print not red;
note right: no color
endif
partition End {
:end;
}
-> this is the end;
end
@enduml
```

With style

You can use [style](#) to change rendering of elements.

-
-

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

```

<style>
activityDiagram {
  BackgroundColor #33668E
  BorderColor #33668E
  FontColor #888
  FontName arial

  diamond {
    BackgroundColor #ccf
    LineColor #00FF00
    FontColor green
    FontName arial
    FontSize 15
  }
  arrow {
    FontColor gold
    FontName arial
    FontSize 15
  }
  partition {
    LineColor red
    FontColor green
    RoundCorner 10
    BackgroundColor PeachPuff
  }
  note {
    FontColor Blue
    LineColor Navy
    BackgroundColor #ccf
  }
}
document {
  BackgroundColor transparent
}
</style>
start
:init;
-> test of color;
if (color?) is (<color:red>red) then
:print red;
else
:print not red;
note right: no color
endif
partition End {
:end;
}
-> this is the end;
end
@enduml

```

