

- 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 > Component Diagram >

sponsors 20 patreon 122 liberapay 10 paypal 296

❏ Диаграмма компонентов

Let's have few examples : Let's have few examples.

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

⏏ ❏ Компоненты

Обозначения компонентов должны быть заключены в квадратные скобки.

Также можно использовать ключевое слово `component` для объявления компонента. Вы можете объявить алиас с помощью ключевого слова `as` . Этот алиас может быть использован позже, при объявлении связей.

@startuml

[First component]
[Another component] as Comp2
component Comp3
component [Last\ncomponent] as Comp4

@enduml

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

⏏ ❏ Интерфейсы

Для обозначения интерфейса используется символ `()` (потому что он выглядит как круг).

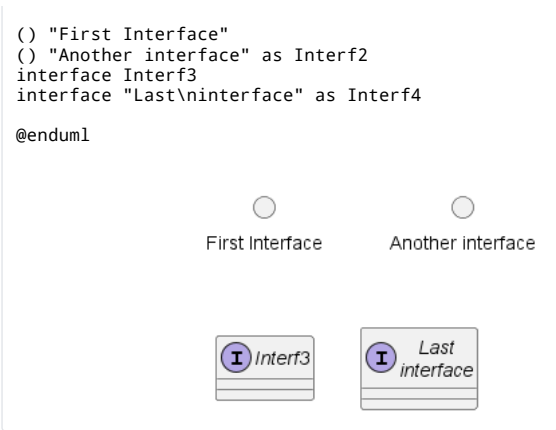
Также возможно использование ключевого слова `interface` для объявления интерфейса. Вы можете объявить алиас с помощью ключевого слова `as` . Этот алиас может быть использован позднее, когда будут задаваться связи.

Далее мы увидим, что задание интерфейсов опционально.



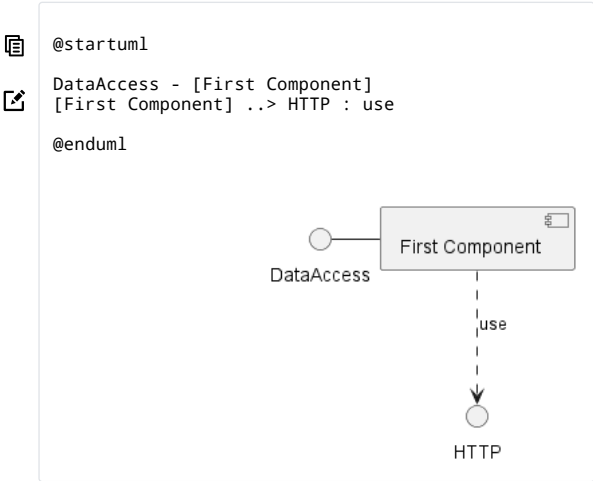
- Компоненты
- Интерфейсы
- Простой пример
- Использование заметок
- Группирование компонентов
- Изменение направления стрелок
- Use UML2 notation
- Использовании нотации UML1
- Use rectangle notation (remove UML notation)
- Длинное описание
- Индивидуальные цвета
- Использование Sprite в стереотипах
- Skinparam
- Specific SkinParameter
- Hide or Remove unlinked component
- Hide, Remove or Restore tagged component or wildcard
- Display JSON Data on Component diagram
- Port [port, portIn, portOut]

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



Простой пример

Отношения между элементами создаются с помощью комбинации точечных линий (. .), прямых линий (--) и стрелок (-->).



Использование заметок

Вы можете использовать ключевые слова `note left of` , `note right of` , `note top of` , `note bottom of` чтобы задать метки, относящиеся к одному объекту.

Заметка также может быть задана не прикреплённой, используя ключевое слово `note` , а затем прикреплена к другим объектам, используя символ `...` .

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

```
interface "Data Access" as DA
DA - [First Component]
[First Component] ..> HTTP : use
note left of HTTP : Web Service only
note right of [First Component]
  A note can also
  be on several lines
end note
@enduml
```

```
graph TD
    DA((Data Access)) --- FC[First Component]
    FC -.->|use| HTTP((HTTP))
    HTTP --- WS[Web Service only]
    FC --- Note1[A note can also be on several lines]
```

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

Группирование компонентов

Вы можете использовать несколько ключевых слов `package` , чтобы группировать компоненты и интерфейсы вместе.

- package
- node
- folder
- frame
- cloud
- database



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

```
package "Some Group" {
    HTTP - [First Component]
    [Another Component]
}

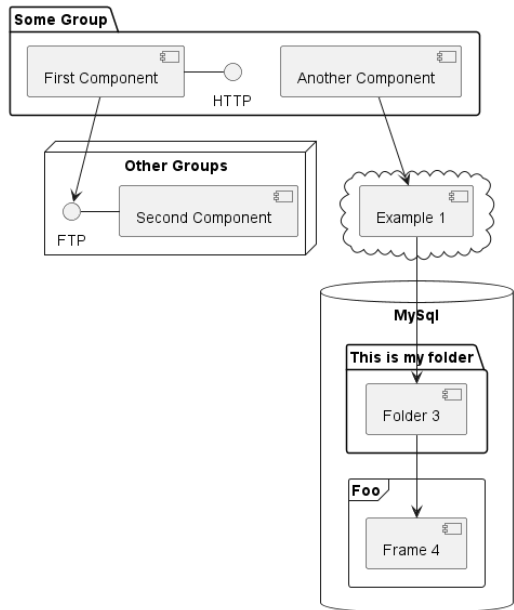
node "Other Groups" {
    FTP - [Second Component]
    [First Component] --> FTP
}

cloud {
    [Example 1]
}

database "MySQL" {
    folder "This is my folder" {
        [Folder 3]
    }
    frame "Foo" {
        [Frame 4]
    }
}

[Another Component] --> [Example 1]
[Example 1] --> [Folder 3]
[Folder 3] --> [Frame 4]

@enduml
```



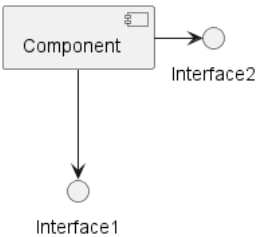
Изменение направления стрелок

По умолчанию, связи между классами имеют два типа -- и ориентированы вертикально. Можно создавать горизонтальные связи с помощью одного типа (или точки), вот так:



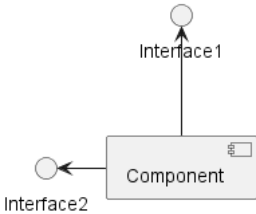
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
[Component] --> Interface1
[Component] -> Interface2
@enduml
```



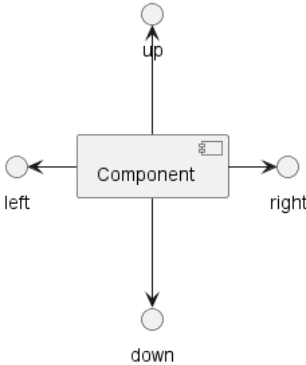
Вы также можете изменять направления, перевернув связь:

```
@startuml
Interface1 <-- [Component]
Interface2 <- [Component]
@enduml
```



Также, можно изменить направление стрелки добавлением ключевых слов left , right , up или down внутри стрелки:

```
@startuml
[Component] -left-> left
[Component] -right-> right
[Component] -up-> up
[Component] -down-> down
@enduml
```



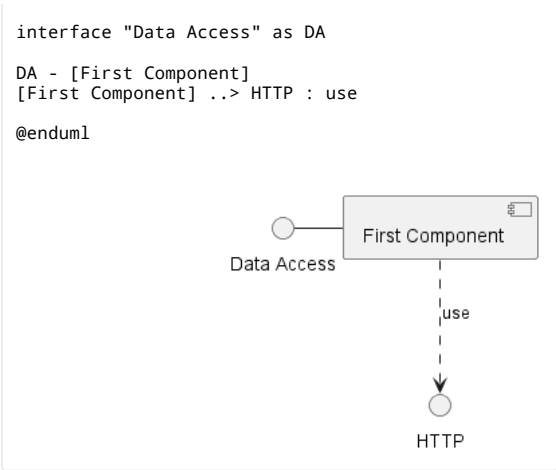
Вы можете сократить запись, используя только первую букву направления (например, -d- вместо -down-) или две первые буквы (-do-).

Пожалуйста, заметьте, что не стоит использовать эту функциональность без особой надобности: *Graphviz* обычно даёт хорошие результаты без дополнительной настройки.

Use UML2 notation

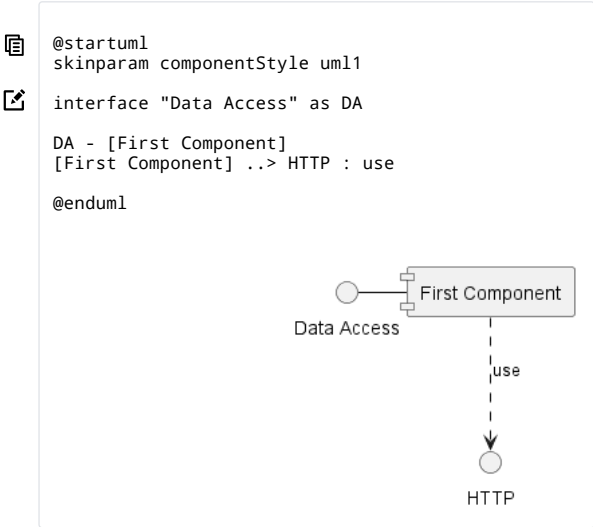
By default (from v1.2020.13-14), UML2 notation is used.

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



Использовании нотации UML1

Команда `skinparam componentStyle uml1` используется, чтобы переключиться на нотацию UML1.



Use rectangle notation (remove UML notation)

The `skinparam componentStyle rectangle` command is used to switch to rectangle notation (without any UML notation).

-
-

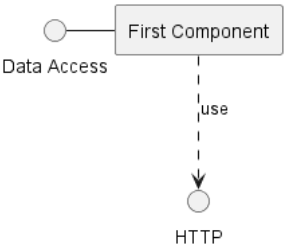
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam componentStyle rectangle

interface "Data Access" as DA

DA - [First Component]
[First Component] ..> HTTP : use

@enduml
```



Длинное описание

Для помещения многострочного текста в тело компонента используются квадратные скобки

```
@startuml
component comp1 [
This component
has a long comment
on several lines
]
@enduml
```

This component
has a long comment
on several lines

Индивидуальные цвета

Вы можете задать цвет после определения компонента.

```
@startuml
component [Web Server] #Yellow
@enduml
```

Web Server

Использование Sprite в стереотипах

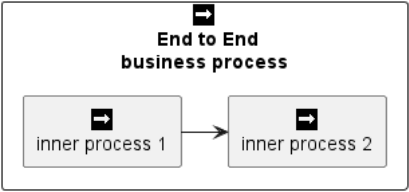
Можно использовать спрайты внутри компонентов стереотипа.

-
-

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

```
sprite $businessProcess [16x16/16] {
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFF0FFFFF
  FFFFFFFFF0FFFFF
  FF0000000000FFF
  FF0000000000FFF
  FF0000000000FFF
  FFFFFFFFF0FFFFF
  FFFFFFFFF0FFFFF
  FFFFFFFFF0FFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
  FFFFFFFFFFFFFFFF
}

rectangle " End to End\nbusiness process" <<$busi
rectangle "inner process 1" <<$businessProcess>>
rectangle "inner process 2" <<$businessProcess>>
src -> tgt
}
@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 задать цвет и шрифт для компонентов и интерфейсов с заданными шаблонами.



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

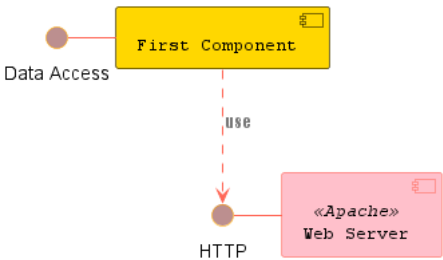
```
skinparam interface {
  backgroundColor RosyBrown
  borderColor orange
}

skinparam component {
  FontSize 13
  BackgroundColor<<Apache>> Pink
  BorderColor<<Apache>> #FF6655
  FontName Courier
  BorderColor black
  BackgroundColor gold
  ArrowFontName Impact
  ArrowColor #FF6655
  ArrowFontColor #777777
}

() "Data Access" as DA
Component "Web Server" as WS << Apache >>

DA - [First Component]
[First Component] ..> () HTTP : use
HTTP - WS

@enduml
```



```
@startuml

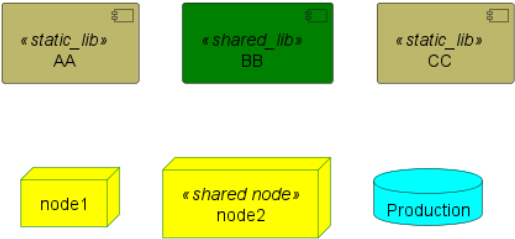
skinparam component {
  backgroundColor<<static_lib>> DarkKhaki
  backgroundColor<<shared_lib>> Green
}

skinparam node {
  borderColor Green
  backgroundColor Yellow
  backgroundColor<<shared_node>> Magenta
}
skinparam databaseBackgroundColor Aqua

[AA] <<static_lib>>
[BB] <<shared_lib>>
[CC] <<static_lib>>

node node1
node node2 <<shared node>>
database Production

@enduml
```



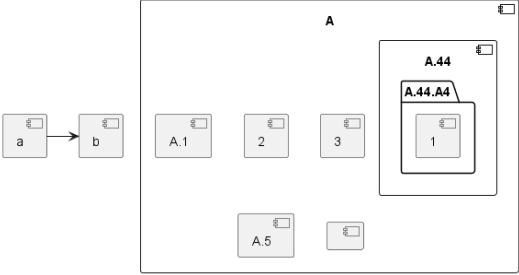
WARNING This translation need to be updated. WARNING

Specific SkinParameter

componentStyle

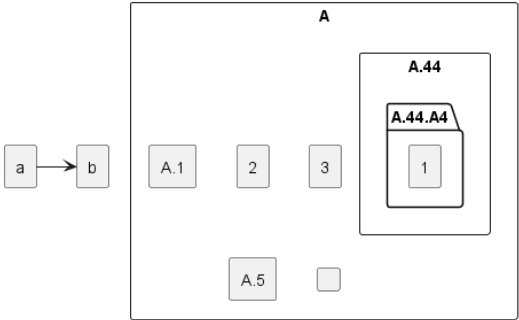
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
@startuml
skinparam BackgroundColor transparent
skinparam componentStyle uml2
component A {
  component "A.1" {
  }
  component A.44 {
    [A4.1]
  }
  component "A.2"
  [A.3]
  component A.5 [
A.5]
  component A.6 [
]
}
[a]->[b]
@enduml
```



- If you want to suppress it, and to have only the rectangle, you can use `skinparam componentStyle rectangle`

```
@startuml
skinparam BackgroundColor transparent
skinparam componentStyle rectangle
component A {
  component "A.1" {
  }
  component A.44 {
    [A4.1]
  }
  component "A.2"
  [A.3]
  component A.5 [
A.5]
  component A.6 [
]
}
[a]->[b]
@enduml
```



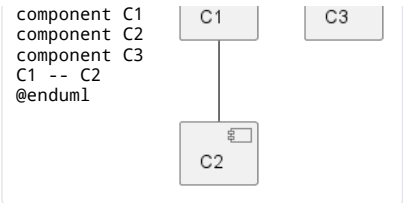
[Ref. 10798]

Hide or Remove unlinked component

By default, all components are displayed:

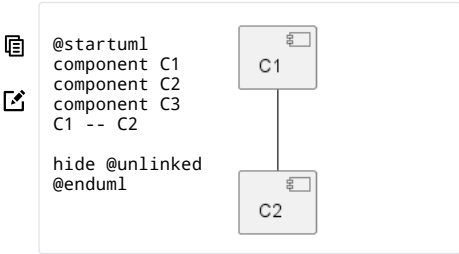
-
-

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

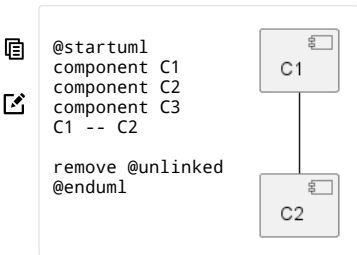


But you can:

- hide @unlinked components:



- or remove @unlinked components:

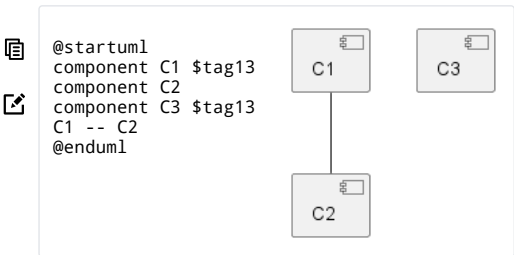


[Ref. [QA-11052](#)]

Hide, Remove or Restore tagged component or wildcard

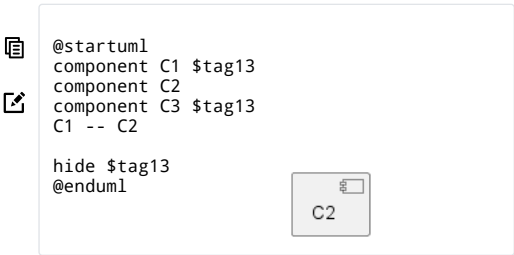
You can put \$tags (using \$) on components, then remove, hide or restore components either individually or by tags.

By default, all components are displayed:



But you can:

- hide \$tag13 components:



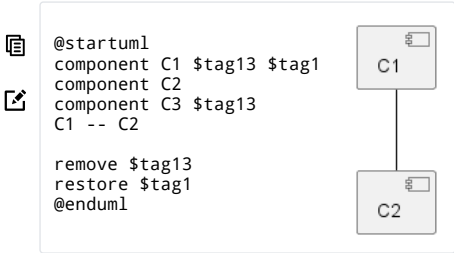
- or remove \$tag13 components:

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

```
component C1 $tag1
component C2
component C3 $tag1
C1 -- C2

remove $tag1
@enduml
```

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



- or remove * and restore \$tag1 components:

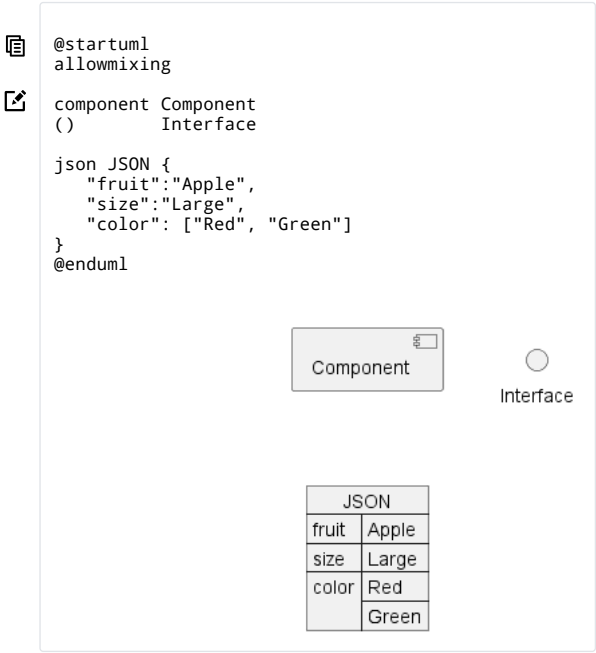
```
@startuml
component C1 $tag13 $tag1
component C2
component C3 $tag13
C1 -- C2

remove *
restore $tag1
@enduml
```

[Ref. [QA-7337](#) and [QA-11052](#)]

Display JSON Data on Component diagram

Simple example



[Ref. [QA-15481](#)]

For another example, see on [JSON](#) page.

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

You can added **port** with `port` , `portin` and `portout` keywords.

Port

```
@startuml
[c]
component C {
  port p1
  port p2
  port p3
  component c1
}

c --> p1
c --> p2
c --> p3
p1 --> c1
p2 --> c1
p2 --> c1
@enduml
```

PortIn

```
@startuml
[c]
component C {
  portin p1
  portin p2
  portin p3
  component c1
}

c --> p1
c --> p2
c --> p3
p1 --> c1
p2 --> c1
p2 --> c1
@enduml
```

PortOut

```
@startuml
component C {
  portout p1
  portout p2
  portout p3
  component c1
}
[o]
p1 --> o
p2 --> o
p3 --> o
c1 --> p1
@enduml
```

Mixing PortIn & PortOut

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

```
[i]
component C {
  portin p1
  portin p2
  portin p3
  portout po1
  portout po2
  portout po3
  component c1
}
[o]

i --> p1
i --> p2
i --> p3
p1 --> c1
p2 --> c1
c1 --> po1
po1 --> o
po2 --> o
po3 --> o
@enduml
```

