PlantUML

AN OPEN SOURCE PROJECT FOCUSED ON UML DIAGRAM DEVELOPMENT

ANDRÉS GARCÍA CELÍS - UO264345

EDUARDO LAMAS SUÁREZ - UO257611

FERNANDO SÁENZ DE SANTA MARÍA MODROÑO - UO266754

SARA VECINO GARCÍA - UO264093

What is it?

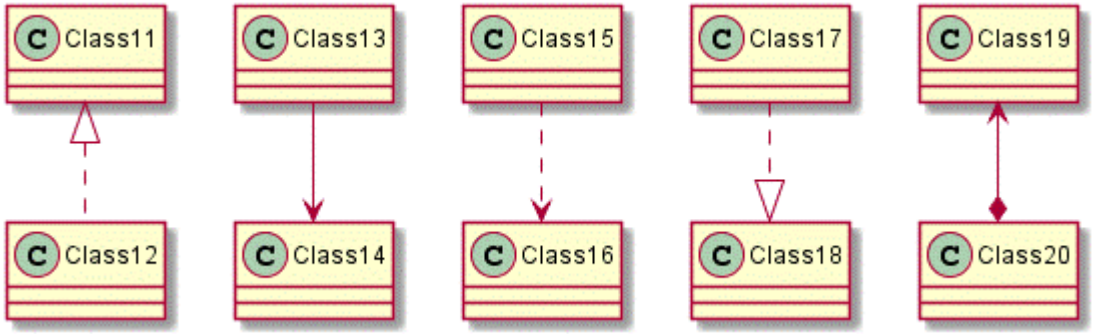
PlantUML is an open-source standalone tool that allows users to create UML diagrams from a plain text file. “Drawing” a diagram with Plant is similar to coding each object and its relation with the other objects. It defines a DSL which has the sole purpose of being human readable.

A large number of applications offer complete or partial integration, such as:

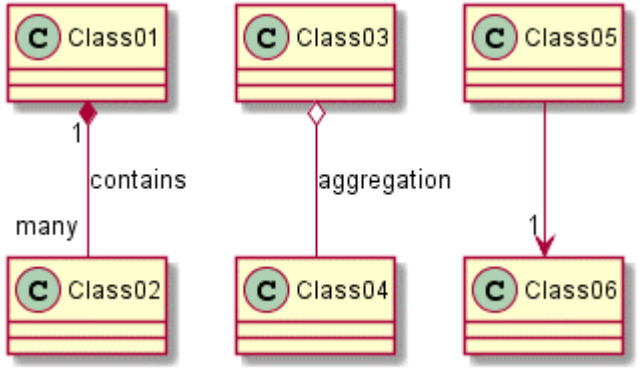
- GitHub
- Eclipse
- Google Docs
- Microsoft Word
- LaTeX
- IntelliJ IDEA
- Atom
- Visual Studio Code

A simple overview

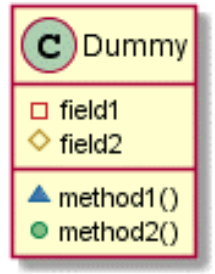
```
@startuml
Class11 <|.. Class12
Class13 --> Class14
Class15 ..> Class16
Class17 ..|> Class18
Class19 <--* Class20
@enduml
```



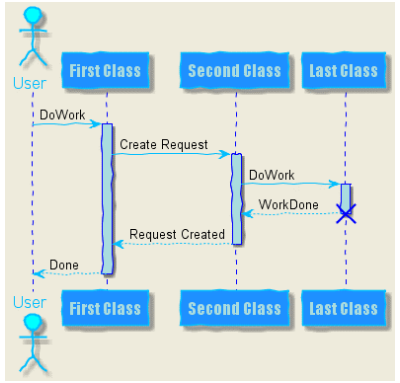

```
@startuml
Class01 "1" *-- "many"
Class02 : contains
Class03 o-- Class04 : aggregation
Class05 --> "1" Class06
@enduml
```



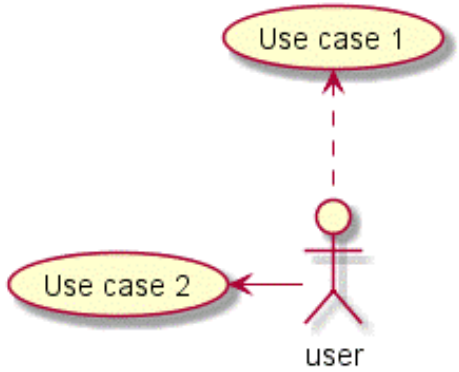

```
@startuml
class Dummy {
-field1
#field2
~method1()
+method2()
}
@enduml
```



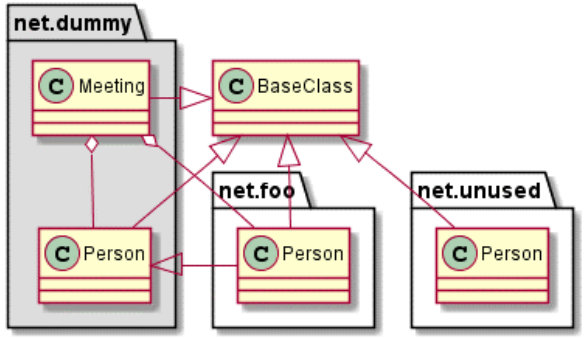
More diagram examples



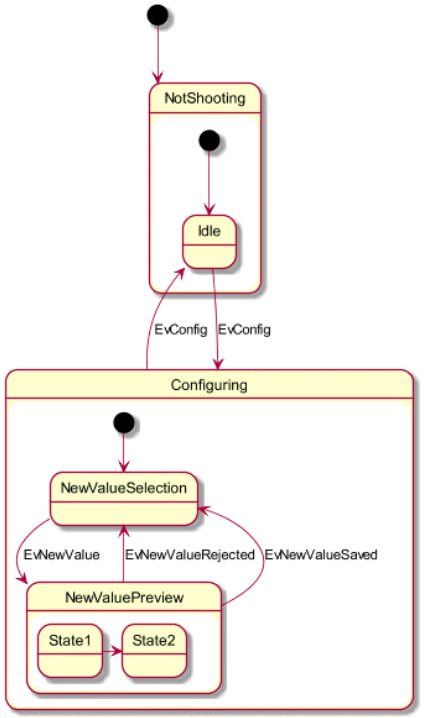
Sequence diagram: Custom rendered



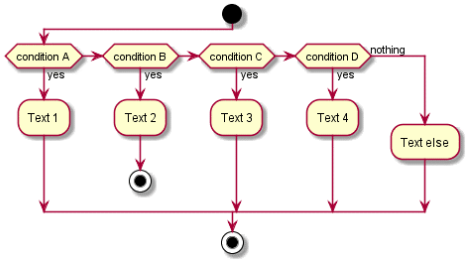
Use case diagram



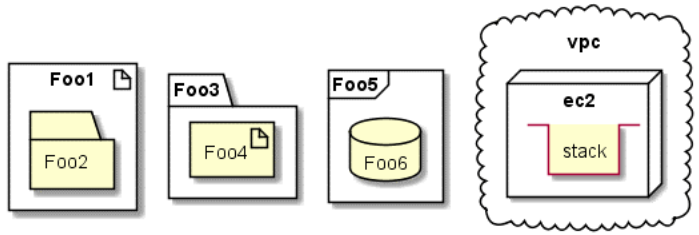
Class diagram



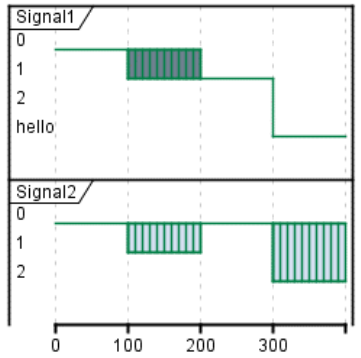
State diagram



Activity diagram (flowchart)



Deployment diagram



Timing diagram

Stakeholders

- Project designers/Software architects
- Software developers
- Blind software developers
- Software students
- Open-source community
- Implementations, extensions and add-ons

Constraints

- Java programming language
- Easily modifiable diagrams
- Graphviz
- Open source project
- Continuous integration
- Pay special attention to the implementation design
- Maintain PlantUML language specification without changes

Quality attributes/goals

- Generating diagrams from simple and human-readable text
- Implement the tool mantaining the PlantUML DSL syntax stable
- Keep the source code for the tool open, under GPL v3.0
- Offer plenty of diagram options to generate

Development aspects

PlantUML is implemented in Java by Arnaud Roques, open-source and continuously integrated.

- Dependent on Graphviz and Java.
- Some of the design patterns used:
  - Command
  - Abstract Factory
  - Template Method
  - Strategy
- Very maintainable and reusable code.

Modules of the project

PlantUML

PlantUML-server

PlantUML-mit

PlantUML-stdlib

Technological environment

Java™

Maven™

SOURCEFORGE

<APACHE ANT>

Travis CI

GitHub

Conclusion

Questions