User Manual

UF-yUML

15 Octobre 2015

Introduction

This document is given you some keys to use the UF-yUML project, and especially to deals with the syntax. The UF-yUML project allows you to draw very easyly UML diagramm, and generate the associate java code, in an Object-Oriented way.

Installation This project is based on yuml 0.1 (https://pypi.python.org/pypi/yuml). In order to use UF-yUML, you will need to install yuml 0.1 (github repository : https://github.com/wandernauta/yuml/). Once you've install yuml 0.1, you only need to clone the UF-yUML in the same folder.

Utilisation First, you need to create a folder to your project.

Then, create in your project folder a class folder name "class". This folder will contains all the informations about your classes (attributes, methods), with one file for one class. Section 2 shows you how to fill a class file with all its informations.

The next step is to build liaisons between your classes. Create one or more file in your folder project whose name begins by "liaisons". The section 3 shows you how to fill a liaisons file.

After running (see Section 1 dealing with the run commands), UF-yUML will create a file diagramm.*** in your folder path with your project folder, and create a folder named "java" with your java code if needed (see Section 3).

1 Run commands

The typical run command is :

```
python uf-yuml.py -f folder/
```

Options :

Only the -f is required

-f the name of your project folder

-o the output format: jpg, png or pdf (default: png).

-s the UML output shape: scruffy, nofunky or plain (default: plain).

-j the java code generator: True or False (default False)

-g automatics getter and setter: True are False (default False).

-m displays the methods in the UML diagramm: True or False (default False).

2 Classes (or Interface, etc...)

One class file is divided into three parts : the name, the attributs and the methods.

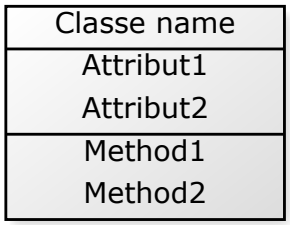
The following class file :

```
Classe name

Attribut1
Attribut2

Method1
Method2
```

will gave the following UML result :

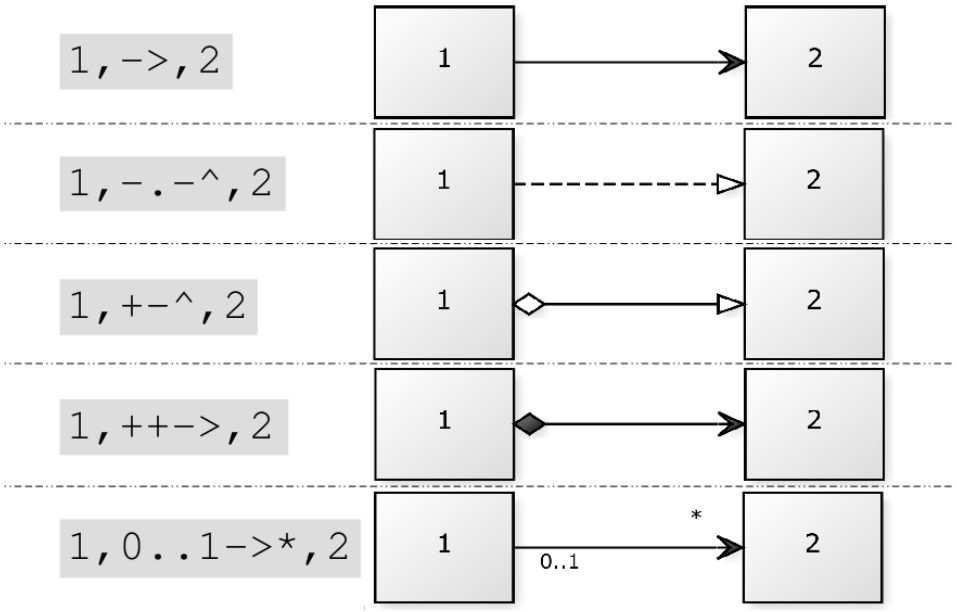


3 Liaisons

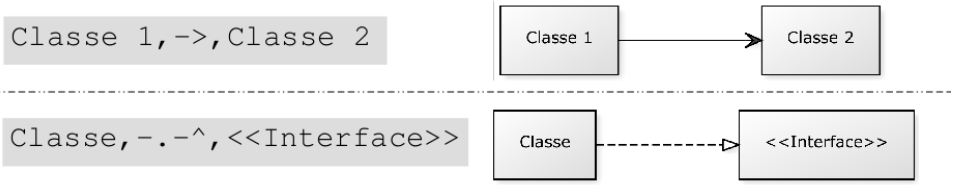
Each row of a liaisons file corresponf to a liaison. The format of a row is :

```
name_1,-,name_2
```

name_1 and name_2 are the names of the two class you want to relie. - correspond to a simple line without arrow. You can change this string to change the shape of your line. Here are somes examples:



You can also use space for your classes names, or use « for an interface:



Java generator

If you want to generate java code, you have to respect some syntaxic rules about class files. In particularly:

- Interfaces format must be «name».
- Interfaces implementations must be like −.−^
- Classes heritage must be like −^
- Attributs format must be like "name: type"
- Methods format must be like "name(arguments): type"
- Public attributs or methods mut begin by "+ "
- Private attributs or methods mut begin by "- "