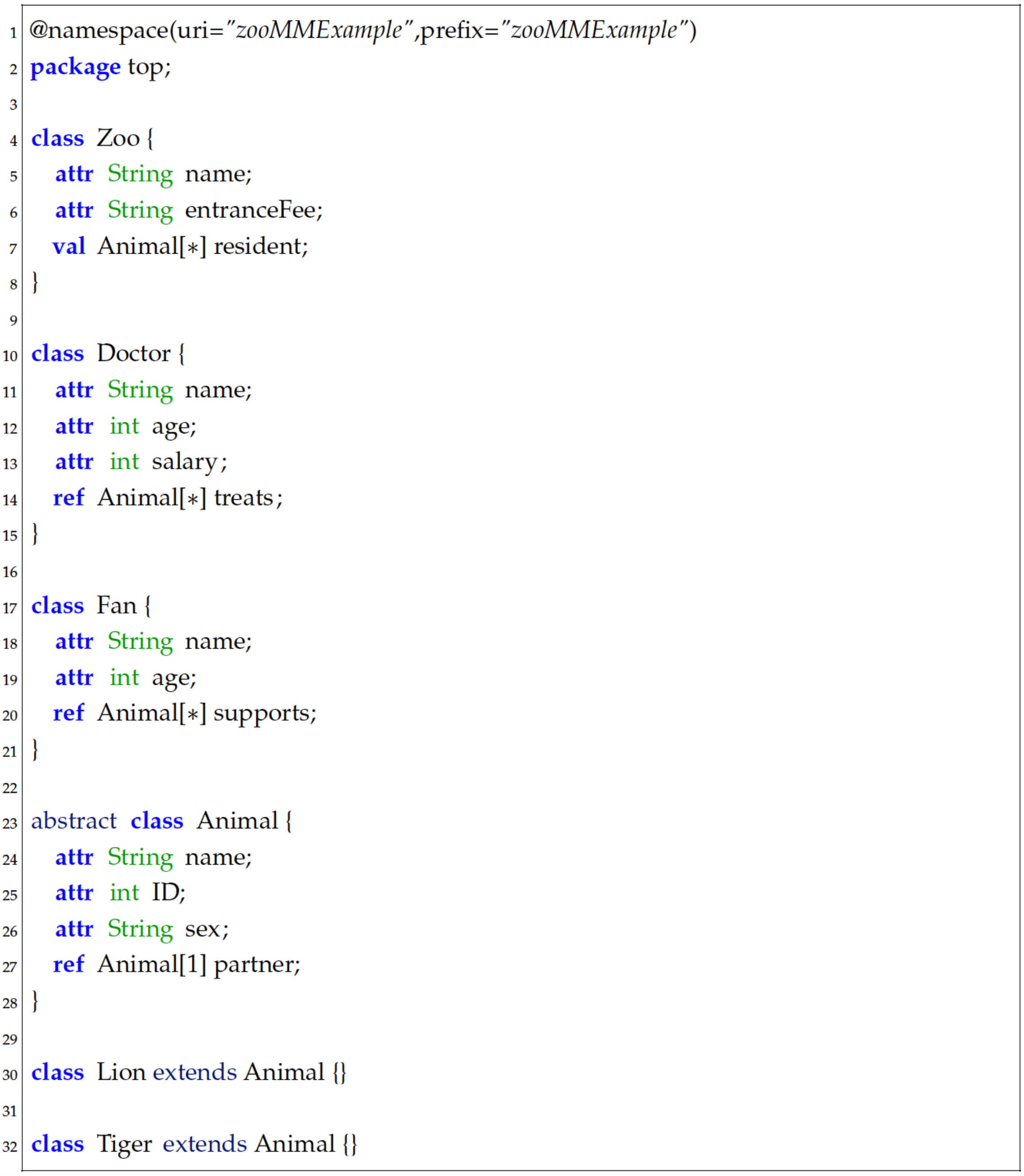
Jorvik Task: Default

This is a little bit of background knowledge needed to complete this task.

The Eclipse Modelling Framework (EMF) is probably the most widely used metamodelling infrastructure and is part of the Eclipse Foundation. It aligns with the four-layer MOF architecture and provides modellers with a metamodelling language called *Ecore*. The *Emfatic* textual language can be used to define Ecore metamodels in EMF. An example of Emfatic code for defining a metamodel is presented below.



```
@namespace(uri="zooMMExample",prefix="zooMMExample")
package top;

class Zoo {
  attr String name;
  attr String entranceFee;
  val Animal[*] resident;
}

class Doctor {
  attr String name;
  attr int age;
  attr int salary;
  ref Animal[*] treats;
}

class Fan {
  attr String name;
  attr int age;
  ref Animal[*] supports;
}

abstract class Animal {
  attr String name;
  attr int ID;
  attr String sex;
  ref Animal[1] partner;
}

class Lion extends Animal {}

class Tiger extends Animal {}
```

Metamodels defined in Emfatic can be transformed to Ecore automatically (right-click on the Emfatic file and then select generate Ecore).

Ecore allows the definition of *Annotations* attached to any element in the metamodel. An example of an annotation is the one shown in line 1 of the figure above.

Jorvik is a tool that is able to generate Papyrus editors for UML profiles, like those you created in the first part of the experiment, automatically using as input annotated Ecore metamodels.

A number of annotations you can use in Jorvik are:

@Diagram on root EPackage. Its parameters include

- name: to defined the name of the diagrams as they will appear on Papyrus menus (usually, this is the name of your DSL)
- icon: the icon that will be used for all the diagrams you create using the custom editor you build (**note**: icon images should be stored under the *icons* folder)

Example: @Diagram(name=”Zoo”, icon=”icons/zoo.png”)

@Node on meta-elements. Its parameters include

- base: to specify what UML meta-class you are basing this annotation on, e.g. base=”Class”

- shape: to specify the location of the svg file you’d like adhere to your Stereotype in the UML profile editor (**note**: shape images should be stored under the *shapes* folder)

- icon: to specify the icon of the tool in the palette (**note**: icon images should be stored under the *icons* folder)

- bold: to specify if the label font should be in bold (i.e. “true”/”false”)

- fontHeight: to specify the font size of the label (e.g. “15”)

@Edge on either references or meta-elements, its parameters include

- base: to specify what UML meta-class you are basing this annotation on, e.g. base=”Association”
- source: the reference name of the source of the edge (if you apply the @Edge annotation on a class)
- target: the reference name of the target of the edge (if you apply the @Edge annotation on a class)

- icon: to specify the icon of the tool in the palette

- bold: to specify if the label font should be in bold (i.e. “true”/”false”)

- fontHeight: to specify the font size of the label

- lineStyle: to specify the style of the line (e.g. “dotted”/”dashed”)

In this task you are required to create the equivalent annotated Ecore metamodels (or an Emfatic file if you prefer) for the profiles we provide and use Jorvik to generate the UML Profile editor for them.

Subtasks:

1. Define the annotated Ecore metamodel for the DSL provided to you. Start with defining one Node and one Edge in the UML profile.
    a. Remember that you can always check your solution by trying to generate the editor and see the results/feedback. NB.: In order to do that for a partial version of the profile you need to have an Ecore/Emfatic that has no errors thus you might need to write empty classes for the references included in the 2 stereotypes that you picked to implement in this task.
    b. Note: In UML, classes in italics denote *abstract* classes.
2. For the each scenario, associate the elements with their corresponding SVG shapes and palette icons
    a. In part 1 (where you work on the Webpage scenario) make the font of all the stereotypes bold.
    b. In part 2 (where you work on the FTA scenario) make the font of all the stereotypes bold and set the font height to 18.
3. Complete the whole metamodel.
    a. NB.: For the FTA example, the class FaultTree should not be annotated as we don't want to create a graphical element for it.
4. Use Jorvik to generate your editor.

Note: You have 50 minutes to complete this task. You will receive essential information after 30 minutes (if needed).