1. Models: Objects and relations defined in library (possible to create separate files?)
2. Program in Python should permit to:
   a) Parse a textual object file (.rau):
      Read an object file (a main object file) and construct an object type in Python. The program can check the grammar of the object description and signal errors if there is any.
   b) Modify the object and save the modifications:
      The program should permit the user to modify an object, that is, to add, remove or change something of its domains and then to choose to save the modifications or not to the object files.
   c) Represent the object textually or graphically:
      Once the object is constructed in Python, the program can export a readable textual description or a clear graphical representation of the object.
   d) Abstract an object:
      Once the object is constructed, the program can abstract the object to a level specified by the user. The abstraction can be to remove the properties and to hide the relations to only mention the number of relations to give a succinct description.
   e) Compare different objects:
      The comparison can be that of number of sub-objects and number of relations. For objects of the same “extends”, their properties can be compared. A weight can be assigned to each criterion to give a distance. The differences between different objects can be printed out.
   f) Flatten an object:
      The program permits to flatten an object to a certain level specified by the client. To flatten an object is to export the description of the object in a serial way.
3. Manual:
   a) An exhaustive list of deliverables. Models, programs, manuals…
   b) An exhaustive description of APIs of Rauzy language: name of the function, variables, returned values, errors explanation, tips of usage.
   c) A tutorial for beginners with examples.