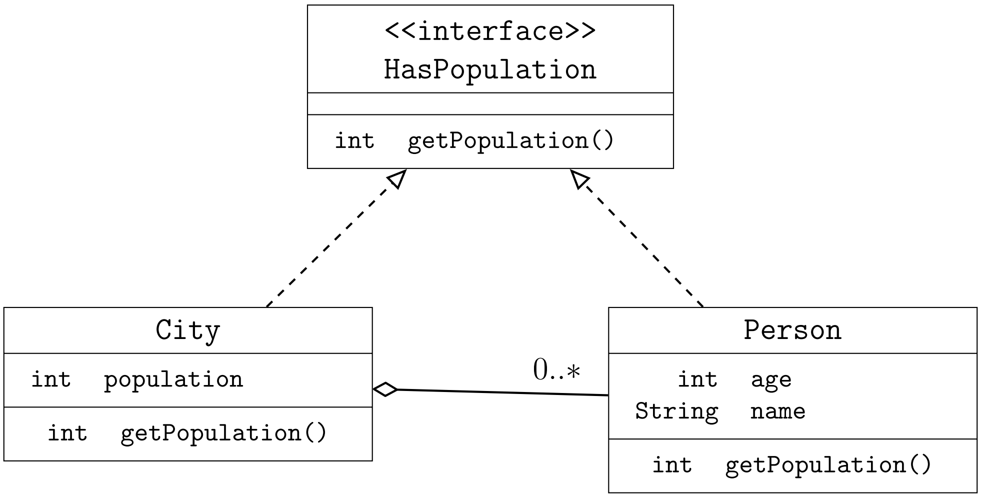
Software and Programming II (SP2) — Lab sheet 11

Object-Oriented Design

2018/19

1. Consider the following UML class diagram.



   Write Java classes that correspond to the above classes and interfaces as well as the relations between them. Instead of method bodies in the classes, just write `{ ... }`.

   - For the access modifiers of the instance variables and the methods, **use the principles of information hiding**.
   - You do not need to provide any constructors.
   - You do not need to add code for instance variables or methods that are not in the diagram.

2. We wish to create an inheritance hierarchy for railway rolling stock. An initial analysis has gathered the following information.

   - For any railway rolling stock, its track gauge in mm is important.
   - An electric locomotive is a piece of railway rolling stock with a certain power in kW and with a certain voltage in Volts.
   - A steam locomotive is a piece of railway rolling stock with a certain power in kW whose boiler has a maximum pressure in Pascal.
   - A carriage is a piece of railway rolling stock with a certain number of seats.
   - A train is railway rolling stock which is composed of a locomotive and one or more carriages.

Devise a suitable inheritance hierarchy for the different kinds of railway rolling stock listed above. Common features should be represented in (possibly abstract) superclasses. Use the UML class diagram notation from the lecture to draw your design.

For each class, in addition to a possible classifier whether it is abstract, only mention the data type and the name of the attributes. You do not need to mention any getters and setters for attributes, nor methods in a class that are already present in an interface that the class implements. For each interface, only mention its name and for each of its methods the name, input types, and output type.

For an aggregation between classes, also give the corresponding multiplicity.