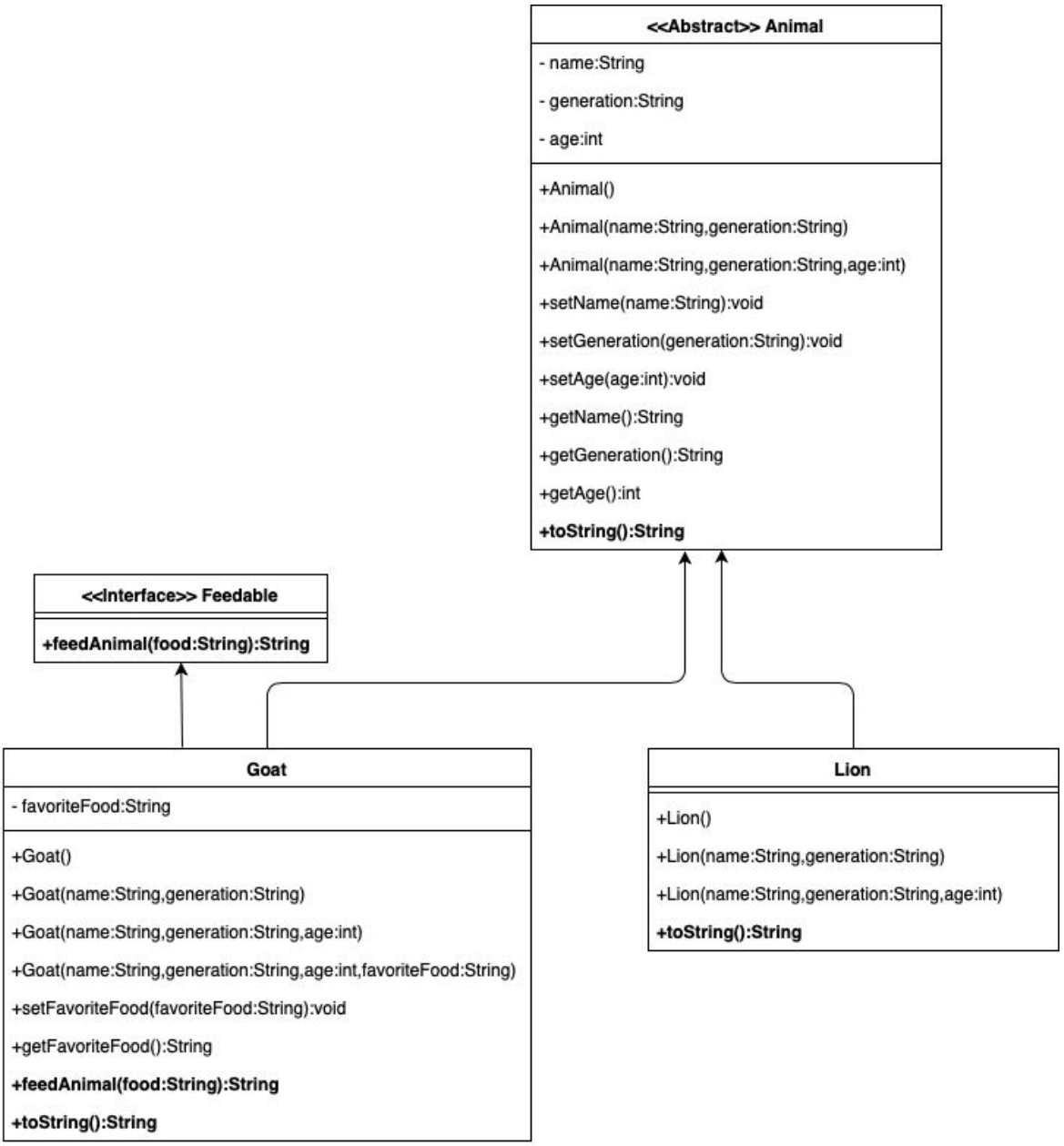
Inheritance - 25 points

- Implement an abstract class Animal with three private attributes **name**, **generation** and **age** including all getter and setter methods. It also has one abstract method **toString()**; (5 points)
- Furthermore, implement an interface **Feedable** that contains only one method **String feedAnimal(String food)**. (5 points)
- Implement the class **Goat** and **Lion** with all getter and setter methods. (5 points)
- Both classes Goat and Lion **inherit** from the abstract class Animal.
- The class Goat has an **additional attribute favoriteFood**, which represents the favorite food of the animal.
- In addition, the class Goat **implements** the interface **Feedable**.
- Calling the feedAnimal of the Goat must return **if it was a good or bad reaction based on the food given** (equals check for the String).
- The **toString** method of each class must return all attributes of the class in text format.

For a Lion, it returns the following string: My name is name, generation of his/her name, and I am age.
For the Goat, it must return the following string: My name is name, generation of his/her name, and I am age. I am pleased if you feed me favoriteFood. (5 points)


<<Abstract>> Animal
- name:String
- generation:String
- age:int
+Animal()
+Animal(name:String,generation:String)
+Animal(name:String,generation:String,age:int)
+setName(name:String):void
+setGeneration(generation:String):void
+setAge(age:int):void
+getName():String
+getGeneration():String
+getAge():int
+toString():String
<<Interface>> Feedable
+feedAnimal(food:String):String
Goat
- favoriteFood:String
+Goat()
+Goat(name:String,generation:String)
+Goat(name:String,generation:String,age:int)
+Goat(name:String,generation:String,age:int,favoriteFood:String)
+setFavoriteFood(favoriteFood:String):void
+getFavoriteFood():String
+feedAnimal(food:String):String
+toString():String
Lion
+Lion()
+Lion(name:String,generation:String)
+Lion(name:String,generation:String,age:int)
+toString():String

Aggregation – 20 points

Add a class called Stable which is designed to model a Animals stable. It contains:

- Three private instance variables: location (String, address of the farm), box (String, the box in the farm they are living), and farm (String, the name of the farm); (5 points)
- Add a constructor to initialize the location, box and farm with the given values; (2 points)
- Add a toString that print the stable information in the following format (2 points):

Location: location (e.g., Ekeberg parken)
Box: box (e.g., Horsebox Nr. 5)
Farm: famr (EKT)

- Modify the Animal class from the previous example accordingly to allow each animal to have a stable. Include methods that allow to get and set the stable for an animal. (5 points)
- Implement a test class called TestAnimal. Write the beginning of the program through public static void main. In the test class create two different Animal objects (one Goat and one Lion) with a Stable only for the Goat. Use the different constructors to do that. Both goat and lion should print its toString. Call the feedAnimal of the Goat one time with a food he/she likes and one with food he/she does not like. Print both reactions of the goat. (6 points)