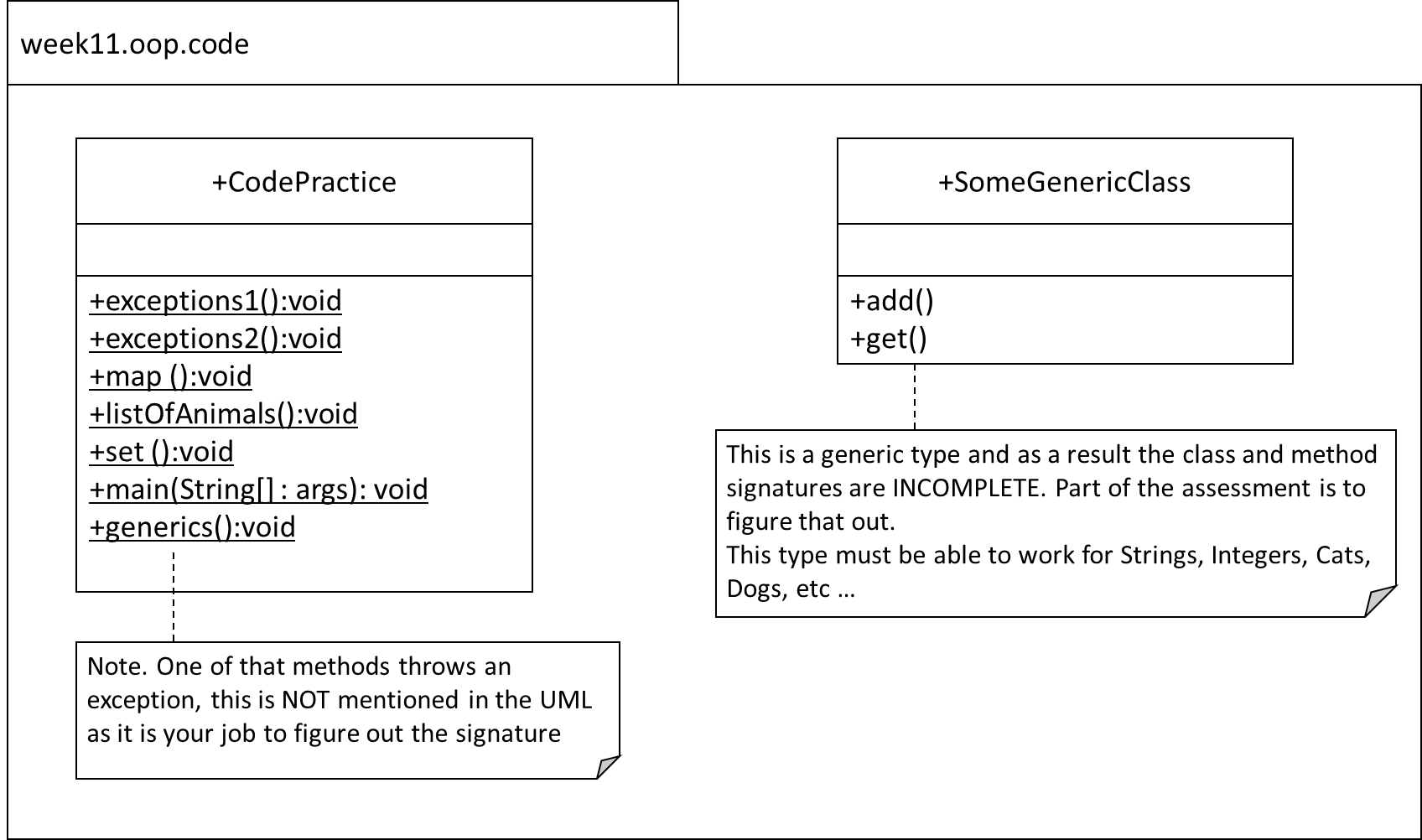
****

Implement the UML and all the following questions in one file.

**Q1.**

Code the following:

In main() call the method exceptions1()

In the method exceptions1(), call the method exceptions2().

In the method exceptions2(), throw an IOException. This exception is caught and handled in the method main().

**Q2**

In the map() method, code the following:

* Declare a reference of type Map that is generically typed for keys of type String and values of type Integer. Name the reference ‘map’. The implementation type is able **to sort itself**! Use type inference (the diamond operator).
* Insert, in this order, the following into the map:
  + “Ford” is 2019
  + “Audi” is 2015
  + “Polo” is 2023
  + “Beetle” is 2012
* Output the map. You should get: {Audi=2015, Beetle=2012, Ford=2019, Polo=2023}
  + note that the map is sorted automatically by its natural ordering of keys i.e. alphabetic order for the names.
* Using the API, output whether the map contains “Ford” as a key. This should output true.
* Using the API, output whether the map contains 2023 as a value. This should output true also.
* Using the API, remove “Polo” from the map.
* Using the API, change “Audi” year to 2016.
* Output the map. You should get : {Audi=2016, Beetle=2012, Ford=2019}
* Using the API and an enhanced-for loop, output the keys in the map. You should get Audi, Beetle, Ford in that order.
* Using the API and an enhanced-for loop, output the values in the map. You should get 2016, 2012, 2019 in that order.

**Q3.**

Code the listOfAnimals method as follows:

* Declare a reference of type List that is generically typed for String i.e. it will only hold String references. Name the reference ‘list’. The implementation type is ArrayList. Use type inference (the diamond operator).
* Insert the following in order: “Ape”, “Bee”, “Cat”, “Elephant”,”Fox”
* Using the API i.e. do not hardcode the index, insert “Dog” in its correct position alphabetically. In other words, when you output the list, you should get [“Ape”, “Bee”, “Cat”, “Dog” ,“Elephant”,”Fox”].
* Using the API, output the size of the list. This should be 6.
* Using the API, output whether the list contains “Panda”. This should output false.
* Using the API, delete “Elephant” from the list.
* Output the list. You should get [“Ape”, “Bee”, “Cat”, “Dog” ,”Fox”]
* Using the API, output if the list is empty or not. This should output false.
* Using the API, change the “Fox” to a “Cat”.
* Output the list. You should get [“Ape”, “Bee”, “Cat”, “Dog” ,”Cat”]

**Q4**

Code the following in the method set():

* Declare a reference of type Set that is generically typed for String i.e. it will only hold **String** references. Name the reference ‘set’. The implementation type is able to sort itself! Use type inference (the diamond operator).
* Insert the following in this order: “10”, “20”, “30”, “40”.
* Output the set. You should get [10,20,30,40].
* Using the API, insert “60”.
* Using the API, delete “30” from the set.
* Output the set. You should get [10,20,40,60].
* Using the API, output if the set contains the string “50”. This should output false.
* Using the API, remove all entries.
* Using the API, output if the set is empty. This should output true.

**Q5.**

Code the following:

Create a generic SomeGenericClass type. After creating the SomeGenericClass type, demonstrate the generic in action by coding the following in the generics() method:

* create the required instances:
  + one that is typed for String’s, called genS
  + the other is typed for Integer’s, called genI
* using genS insert “Holiday” (see the UML for the method name to use).
* Using genS retrieve the String you just inserted and output it to the screen (see the UML for the method name to use).
* using genI insert 12 (see the UML for the method name to use).
* using genI retrieve the Integer you just inserted and output it to the screen (see the UML for the method name to use).