Programming Technologies, Practice Test 1

The following class is given:

```
public class Animal
{
      protected String species;
      protected double age;
      protected double weight;
}
```

1. Extend the class with a constructor. With this constructor all three fields of the class can be given an initial value.

2. Write getter and setter methods for the three fields!

3. Override the **`toString()`** method inherited from the **`Object class`** in a way that it returns all three fields of the animal in the format: `"<species>: <age> years, <weight> kg"` (e.g. `"elephant: 3.5 years, 870 kg"`)!

4. Extend the **`Animal class`** so that its instances can be compared based on their weights.

5. Derive the **`Bird class`** from the **`Animal class.`** Birds should have a logical property called **`canFly`** besides the properties of animals. Add a constructor to the class with which all four fields can be given an initial value.

6. Override the **`toString()`** method inherited from the **`Animal class`** in such a way that all four fields of birds should be returned in a row in the format `"<species>: <age> years, <weight> kg, {can fly|cannot fly}”` (e.g. `"penguin: 5 years, 12 kg, cannot fly"`)!

7. Derive the **`Mammal class`** from the **`Animal class.`** Mammals should have an integer type property called **`numberOfLegs`** besides the properties of animals. Extend the class with a constructor that can set all four fields with an initial value.

8. In the main class of your project, create an Animal type object in the main program, which represents an elephant that is 3.5 years old, and weights 870 kg. Write the object created on the screen!

9. Extend the main program with a part of code which reads 2 birds and 2 mammals from a file (in this particular order, one per line), and places them in a four-element array. The data of the birds and mammals should be separated from each other with spaces. The name of the file is input.txt and you have to create it.

10. Extend the **`Mammal class`** with a method that returns true, if the animal leaves visible footprints in the sand, i.e. if the part of its weight falling to one of its legs exceeds 10 kg.

11. Add an assert to the method above that makes the program stop if the number of legs of the mammal is 0 (e.g. whale). Do not forget to enable assertion for your project.

12. Write a static method in the main class, which receives a collection of animals as a parameter, and returns with the species of the youngest mammal that leaves visible footprints in the sand, and if there is no such mammal, it returns with an empty string.

13. Create a **`Zoo class`**, which can contain an arbitrary number of animals. There should be a possibility of adding new animals, and removing existing animals.

14. Extend the **`Zoo class`** with a method that prints out the animals of the Zoo and the name of the class they are stored in. Also print out the names of all the fields of the containing class.

15. Add a marker annotation type to your project. The name of it should be “NotCheckingValues” this annotation can be used on methods and on constructors and it is visible till runtime. It also appears in the documentation of the classes. Mark all getter methods of the animal class with this annotation.