CSCI 2302
Abstract Classes & Interfaces Chapter Program Assignment

Purpose: Work with class relationships of aggregation and inheritance, abstract classes, interfaces, and abstract methods.

Input: None, hard coded into the program

Output: To the screen

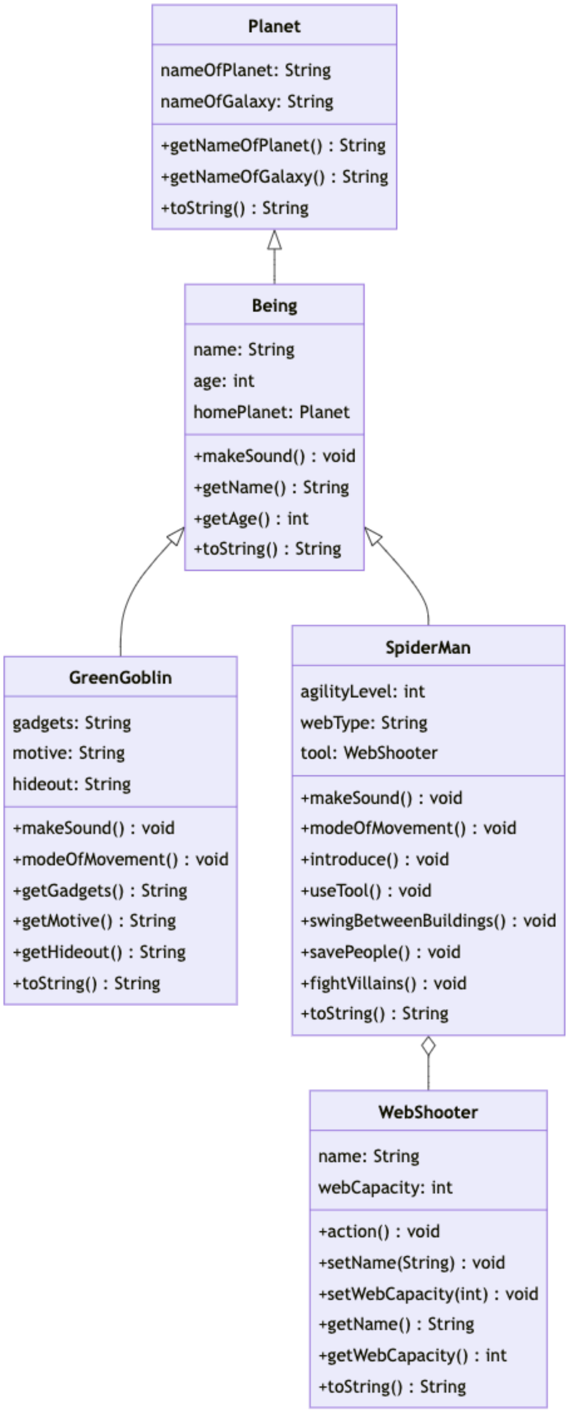
Task:

1. Define/Implement a `Planet` class for representing the planet from which your fictional character (and antagonist) originates from. This class has at least two states/fields/attributes; e.g. `nameOfPlanet` (or `planetName`), `nameOfGalaxy` (or `galaxyName`). `Planet` has at least 4 behaviors; e.g. 1 constructor, 2 getters methods, and `toString` method. Make sure to override the `toString` method from the `Object` class that will provide a description of the planet.
2. Modify the `Being` class to:
    a. be an abstract class.
    b. Have an aggregation relationship – it has-a `Planet`
        i. The `Being` class should have the `Planet` object instantiated in your `Being` class.
    c. have at least 1 abstract method.
3. Implement/Define an interface, `Movement`, with an abstract method of `modeOfMovement.`
4. Modify your fictional character class to inherit from `Being` class.
    a. Modify the constructor to reflect the inheritance from `Being`
    b. Make sure to define the abstract method(s).
        i. Implement/Define the abstract method from the `Being` class
        ii. Implement/Define the abstract method `modeOfMovement`.
5. Implement/Define `YourSFAusername_ACI_Program.java` to do the following:
    a. Instantiate your fictional character.
    b. Invoke the "abstract"/now defined method from the `Being` class that is overridden in your fictional character class.
    c. Invoke the `modeOfMovement` method that is overridden in your fictional character class.

Bonus Points (10 points)

1. Modify your antagonist character class to inherit from `Being` class.
    a. Modify the constructor to reflect the inheritance from `Being`
    b. Make sure to define the abstract method(s).
        i. Implement/Define the abstract method from the `Being` class
        ii. Implement/Define the abstract method `modeOfMovement`.
2. Add to the `YourSFAusername_ACI_Program.java` to do the following:
    a. Instantiate your antagonist character.
    b. Invoke the "abstract"/now defined method from the `Being` class that is overridden in your antagonist character class.
    c. Invoke the `modeOfMovement` method that is overridden in your antagonist character class.

UML DIAGRAM ON NEXT PAGE


Planet
nameOfPlanet: String
nameOfGalaxy: String
+getNameOfPlanet() : String
+getNameOfGalaxy() : String
+toString() : String
Being
name: String
age: int
homePlanet: Planet
+makeSound() : void
+getName() : String
+getAge() : int
+toString() : String
GreenGoblin
gadgets: String
motive: String
hideout: String
+makeSound() : void
+modeOfMovement() : void
+getGadgets() : String
+getMotive() : String
+getHideout() : String
+toString() : String
SpiderMan
agilityLevel: int
webType: String
tool: WebShooter
+makeSound() : void
+modeOfMovement() : void
+introduce() : void
+useTool() : void
+swingBetweenBuildings() : void
+savePeople() : void
+fightVillains() : void
+toString() : String
WebShooter
name: String
webCapacity: int
+action() : void
+setName(String) : void
+setWebCapacity(int) : void
+getName() : String
+getWebCapacity() : int
+toString() : String