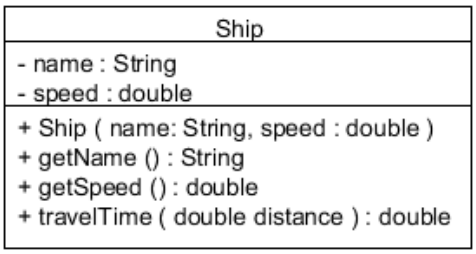
Assignment Ship

CSIS-1400

Learning Objectives:

- Create a class that represents a real life object (in our case a ship)
- Declare getters provide read-access for fields
- Declare a parameterized constructor to initialize the Ship instance (Ship object)
- Develop an algorithm to calculate the desired result
- Use the newly created class as the type of local variables

Description:



Create a project called **AShip** with two java files called **Ship.java** and **ShipApp.java**
Implement the class Ship as specified in the UML class diagram. The method names, parameter lists, and return types need to match the UML class diagram.
No public methods or constructors should be added nor modified.

Output:

```
Cruise Ferry:
Name: Peter Pan
Speed: 21.0 knots
Time to cross bass strait: 15.5 hours

Dinghy:
Name: Laser Dinghy
Speed: 25.0 knots
Time to cross bass strait: 13.0 hours
```

Check List:

- Verify that field and method names are exactly as specified in the UML class diagram
- Make sure the comment with your name is included in both code files
- Watch for proper indentation
- Use empty lines to structure your code by grouping related statements

In Ship.java:

- create a public class called **Ship** (see UML class diagram)
  Include the following three comments that label the three categories of class members

  // fields
  // constructors
  // methods

- Declare two private fields: `name` and `speed`.
  The value of the field speed is the average speed in **knots**.
- Declare a public getter for each of the fields
- Declare a public method called `travelTime`.
  It has one parameter: the distance in miles.
  It returns the number of hours it takes to travel the distance specified in the parameter.
  The value returned should be accurate (not rounded)

  Hint:
  Notice the different units we use: miles, hours and knots.
  You might want to convert the speed from knots to miles per hour in order to calculate the distance.
  Here is an approximation you can use: 1 knot = 1.151 mph

In ShipApp.java

Class `ShipApp` does not represent a real life object. It is not intended to be used as a type of variables.
Class `ShipApp` is a test client. It has the purpose to provide the environment we need to test the class Ship.

Write the following code to test the class Ship:

- Create two instance of type `Ship`.
  The first ship should have the name *Peter Pan* and an average speed of *21* knots.
  It should be assigned to a variable called `cruiseFerry`.
  The second ship should have the name *Laser Dinghy* and an average speed of *25* knots.
  It should be assigned to a variable called `dinghy`.
- Display the information of both ship instances as shown in the output.
  Include a label (e.g. `Cruise Ferry`) to let the user know what information is displayed.
  For each of the ships list the name, speed, and the time it takes to cross Bass Strait between Melbourne (Australia) and Hobart (Tansania) - a distance of 374 miles.
  Display the crossing time **rounded to one decimal place**.

Turning in:

Zip up your project**.** Ensure that your zip file includes both java files and that each java file includes a comment with your name and assignment number. Turn it in via Canvas.