PSA

Lab 2 Assignment 1

Kevin Rodrigues

002615215

Q1.

Problem description:

Designing a MotorBoat class in java, that has capacity, amount in Tank, Max Speed, Current Speed, efficiency and Distance Travelled as the attributes and provide implementation and logic for the methods to: change boat's speed, operate boat for an amount of time at current speed, refueling the boat, find the amount left in the tank, distance travelled

Given: Fuel Used = e * s*s * t,

Where e is efficiency, s is the current speed and t is the time

Distance travelled = s * t

The problem will help to master the logic in java and help in building problems solving and logic building skills

Analysis:

Initially I created the MotorBoat class and defined the instance variables (attributes) as mentioned in the problem. Later, I created the methods as defined in the problem statement. By taking the help of MotorBoatTest.java and the testing output provided, I created the parameterized constructor and wrote programming logic for various methods using the formulas provided and my own logic ability to perform operations.

One difficulty which I faced was in the logic of the last test case. To calculate the distance based on the remaining fuel, I had to consider the time which the boat operates based on the fuel that is remaining in the tank. Eventually, I succeeded in finding the correct logic. Right now, the approach and solution looks good.

Source Code:

```
// MotorBoat.java
package p1.lab2.info6205;
public class MotorBoat {
	private double capacity;
	private double amountInTank;			//amountInTank is the remaining fuel in tank
	private double maxSpeed;
	private double currSpeed;
	private double efficiency;
	private double distance;

	public MotorBoat() {
	}

	public MotorBoat(double capacity, double maxSpeed, double efficiency) {
		super();
		this.capacity = capacity;
		this.maxSpeed = maxSpeed;
		this.efficiency = efficiency;
	}
	public void changeSpeed(double s) {
		if(s>maxSpeed) {
			currSpeed = maxSpeed;
		}
		else {
			currSpeed = s;
		}
	 }
```

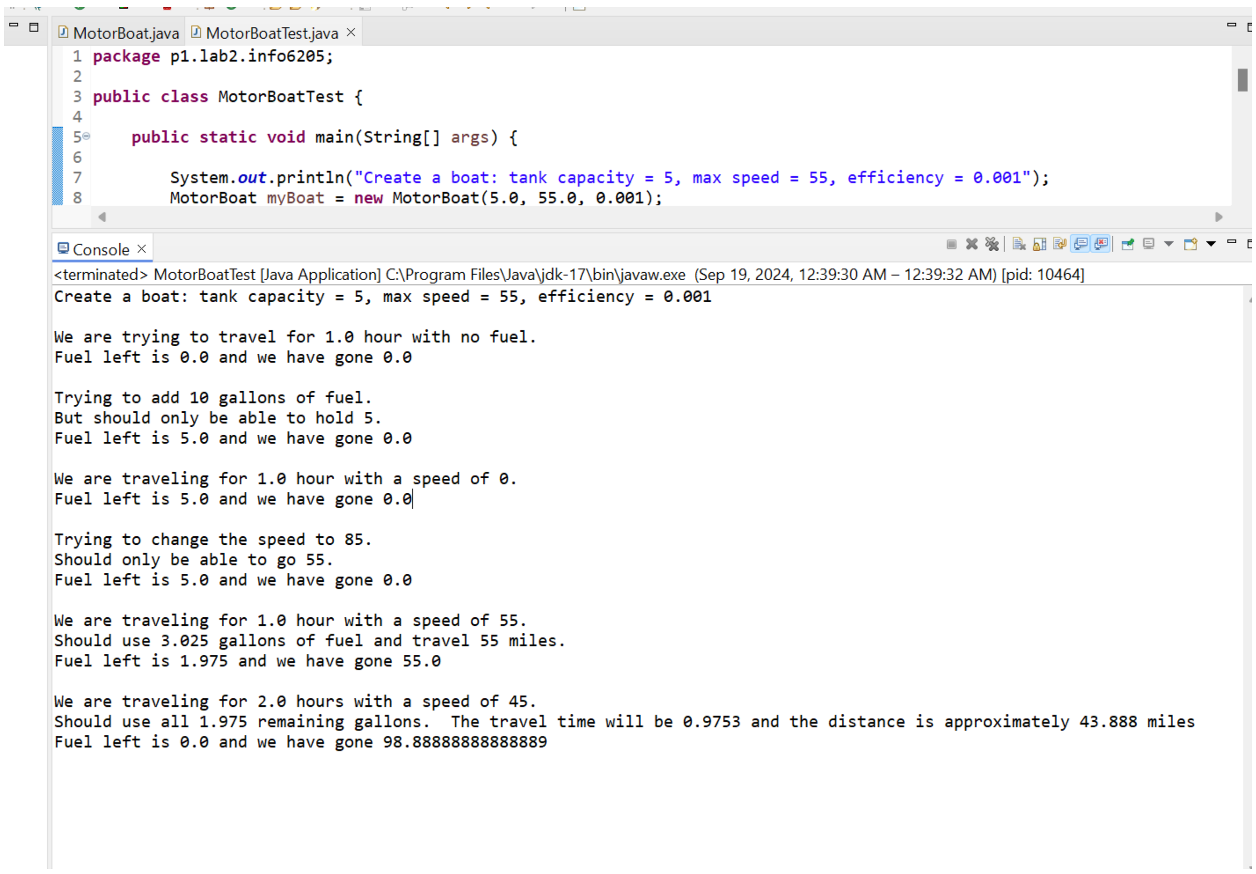
```
public void operateForTime(double time) {
        if (currSpeed == 0) {
                return;
         }
        double fuelUsed = efficiency * currSpeed *currSpeed *time;
        if (amountInTank >= fuelUsed) {
                amountInTank = amountInTank - fuelUsed;
                distance = distance + (currSpeed * time);
        }
        else {
                double operableTimeForBoat = amountInTank / (efficiency * currSpeed *
                currSpeed);
        //      System.out.println("New dist: "+ currSpeed * operableTimeForBoat );
                distance = distance + (currSpeed * operableTimeForBoat);
                amountInTank = 0;
        }
    }


 public void refuelBoat(double d) {
        if(d >= (capacity - amountInTank)) {
                amountInTank = capacity;
        }
        else {
                amountInTank = amountInTank + d;
        }
  }
```

```
        public double fuelRemaining() {
            return amountInTank;
        }

        public double distance() {
            return distance;
        }
    }

// MotorBoatTest.java
package p1.lab2.info6205;

public class MotorBoatTest {
  public static void main(String[] args) {
        System.out.println("Create a boat: tank capacity = 5, max speed = 55, efficiency = 0.001");
        MotorBoat myBoat = new MotorBoat(5.0, 55.0, 0.001);
        System.out.println();

        System.out.println("We are trying to travel for 1.0 hour with no fuel.");
        myBoat.operateForTime(1.0);
        System.out.println("Fuel left is " + myBoat.fuelRemaining()
            + " and we have gone " + myBoat.distance());
        System.out.println();

        System.out.println("Trying to add 10 gallons of fuel.");
        System.out.println("But should only be able to hold 5.");
        myBoat.refuelBoat(10.0);
        System.out.println("Fuel left is " + myBoat.fuelRemaining()
            + " and we have gone " + myBoat.distance());
```

```
System.out.println();

System.out.println("We are traveling for 1.0 hour with a speed of 0.");
myBoat.operateForTime(1.0);
System.out.println("Fuel left is " + myBoat.fuelRemaining()
        + " and we have gone " + myBoat.distance());

System.out.println();

 System.out.println("Trying to change the speed to 85.");
 System.out.println("Should only be able to go 55.");
 myBoat.changeSpeed(85.0);
 System.out.println("Fuel left is " + myBoat.fuelRemaining()
        + " and we have gone " + myBoat.distance());

System.out.println();

System.out.println("We are traveling for 1.0 hour with a speed of 55.");
System.out.println("Should use 3.025 gallons of fuel and travel 55 miles.");
myBoat.operateForTime(1.0);
System.out.println("Fuel left is " + myBoat.fuelRemaining()
        + " and we have gone " + myBoat.distance());

System.out.println();

System.out.println("We are traveling for 2.0 hours with a speed of 45.");

System.out.println("Should use all 1.975 remaining gallons.  "
+ "The travel time will be 0.9753 and the distance is approximately 43.888 miles");
```

```
        myBoat.changeSpeed(45.0);

        myBoat.operateForTime(2.0);

        System.out.println("Fuel left is " + myBoat.fuelRemaining()

                + " and we have gone " + myBoat.distance());

        System.out.println();

    }

}
```

Screenshots/Output:



MotorBoat.java MotorBoatTest.java

```
package p1.lab2.info6205;

public class MotorBoatTest {

    public static void main(String[] args) {

        System.out.println("Create a boat: tank capacity = 5, max speed = 55, efficiency = 0.001");
        MotorBoat myBoat = new MotorBoat(5.0, 55.0, 0.001);
```

Console

```
<terminated> MotorBoatTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Sep 19, 2024, 12:39:30 AM – 12:39:32 AM) [pid: 10464]
Create a boat: tank capacity = 5, max speed = 55, efficiency = 0.001

We are trying to travel for 1.0 hour with no fuel.
Fuel left is 0.0 and we have gone 0.0

Trying to add 10 gallons of fuel.
But should only be able to hold 5.
Fuel left is 5.0 and we have gone 0.0

We are traveling for 1.0 hour with a speed of 0.
Fuel left is 5.0 and we have gone 0.0

Trying to change the speed to 85.
Should only be able to go 55.
Fuel left is 5.0 and we have gone 0.0

We are traveling for 1.0 hour with a speed of 55.
Should use 3.025 gallons of fuel and travel 55 miles.
Fuel left is 1.975 and we have gone 55.0

We are traveling for 2.0 hours with a speed of 45.
Should use all 1.975 remaining gallons.  The travel time will be 0.9753 and the distance is approximately 43.888 miles
Fuel left is 0.0 and we have gone 98.88888888888889
```

Q2.

- ⇨ 2.1

  Problem Description:

  The problem asks us to create a SchoolKid class which should have name, age, teacher's name and greeting as the attributes. Getters and Setters need to be created for the class. We have a testing program sample output and test cases, based upon that we need to create some simple methods and perform basic operations. Namely, we need to display the properties (attributes) of the SchoolKid Object

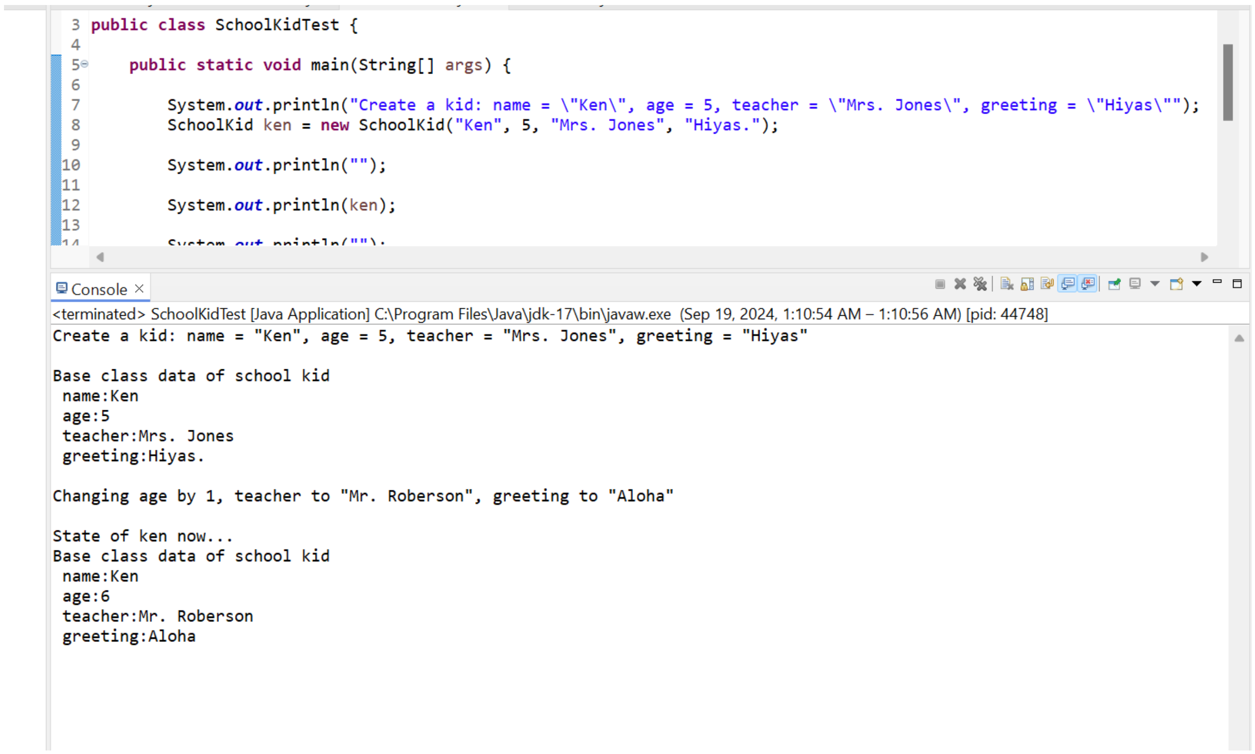
  Analysis:

  I created the SchoolKid class with name, age, teacher, greeting as the instance variables (attributes). I created the getters and setters' methods as suggested. From the test cases I noticed that changing the name of the teacher and changing the greeting is same as setting the name of the teacher and greeting respectively. Hence, I have renamed the setter methods of this attributes to changeTeacher() and changeGreeting() respectively as used in the test case. For Changing the age of the SchoolKid object by 1, I have created a different method named haveBirthday() which increments the age by 1. I have also overridden the toString() method to show the details of the objects.
  The question was simple, and I didn't face any difficulty.
  I feel the solution is already good, but maybe we can have changeTeacher() and changeGreeting() as separate methods and not just setters. Or maybe we have have haveBirthday(){ this.age = this.age+1;}, just to ensure that we are using 'this' keyword and referring to the age variables of the same class.

Source Code:

```
// SchoolKid.java

package p2.lab2.info6205;

public class SchoolKid {

    private String name;
    private int age;
    private String teacher;
    private String greeting;

    public SchoolKid(String name, int age, String teacher, String greeting) {
        super();
        this.name = name;
        this.age = age;
        this.teacher = teacher;
        this.greeting = greeting;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }

    public String getTeacher() {
        return teacher;
    }
    public void changeTeacher(String teacher) {
        this.teacher = teacher;
    }
    public String getGreeting() {
            return greeting;
    }
    public void changeGreeting(String greeting) {
            this.greeting = greeting;
    }

    public int haveBirthday() {
            return age++;
    }

    @Override
    public String toString() {
            return "Base class data of school kid \n name:" + name + "\n age:" + age +
"\n teacher:" + teacher + "\n greeting:" + greeting;
    }

}


// SchoolKidTest.java

package p2.lab2.info6205;

public class SchoolKidTest {
    public static void main(String[] args) {

            System.out.println("Create a kid: name = \"Ken\", age = 5, teacher = \"Mrs.
            Jones\", greeting = \"Hiyas\"");
            SchoolKid ken = new SchoolKid("Ken", 5, "Mrs. Jones", "Hiyas.");

            System.out.println("");

            System.out.println(ken);

             System.out.println("");

            System.out.println("Changing age by 1, teacher to \"Mr. Roberson\", greeting
            to \"Aloha\"");

             System.out.println("");

            ken.haveBirthday();
            ken.changeTeacher("Mr. Roberson");
            ken.changeGreeting("Aloha");
            System.out.println("State of ken now...");
            System.out.println(ken);
        }

    }
```

Screenshots/Output:



```
public class SchoolKidTest {

    public static void main(String[] args) {

        System.out.println("Create a kid: name = \"Ken\", age = 5, teacher = \"Mrs. Jones\", greeting = \"Hiyas\"");
        SchoolKid ken = new SchoolKid("Ken", 5, "Mrs. Jones", "Hiyas.");

        System.out.println("");

        System.out.println(ken);

        System.out.println("");
```

```
<terminated> SchoolKidTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Sep 19, 2024, 1:10:54 AM – 1:10:56 AM) [pid: 44748]
Create a kid: name = "Ken", age = 5, teacher = "Mrs. Jones", greeting = "Hiyas"

Base class data of school kid
 name:Ken
 age:5
 teacher:Mrs. Jones
 greeting:Hiyas.

Changing age by 1, teacher to "Mr. Roberson", greeting to "Aloha"

State of ken now...
Base class data of school kid
 name:Ken
 age:6
 teacher:Mr. Roberson
 greeting:Aloha
```

⇨ 2.2

Problem description:
The problem wants us to derive child class ExaggeratingKid from SchoolKid. We need to override the getAge() method of parent class and change it's implementation to return the actual age plus 2. Also, we need to override the getter method for greeting ie getGreeting() and append "I am the best" string to the end of it. Thus, we need to override and provide customer implementation for these 2 methods.

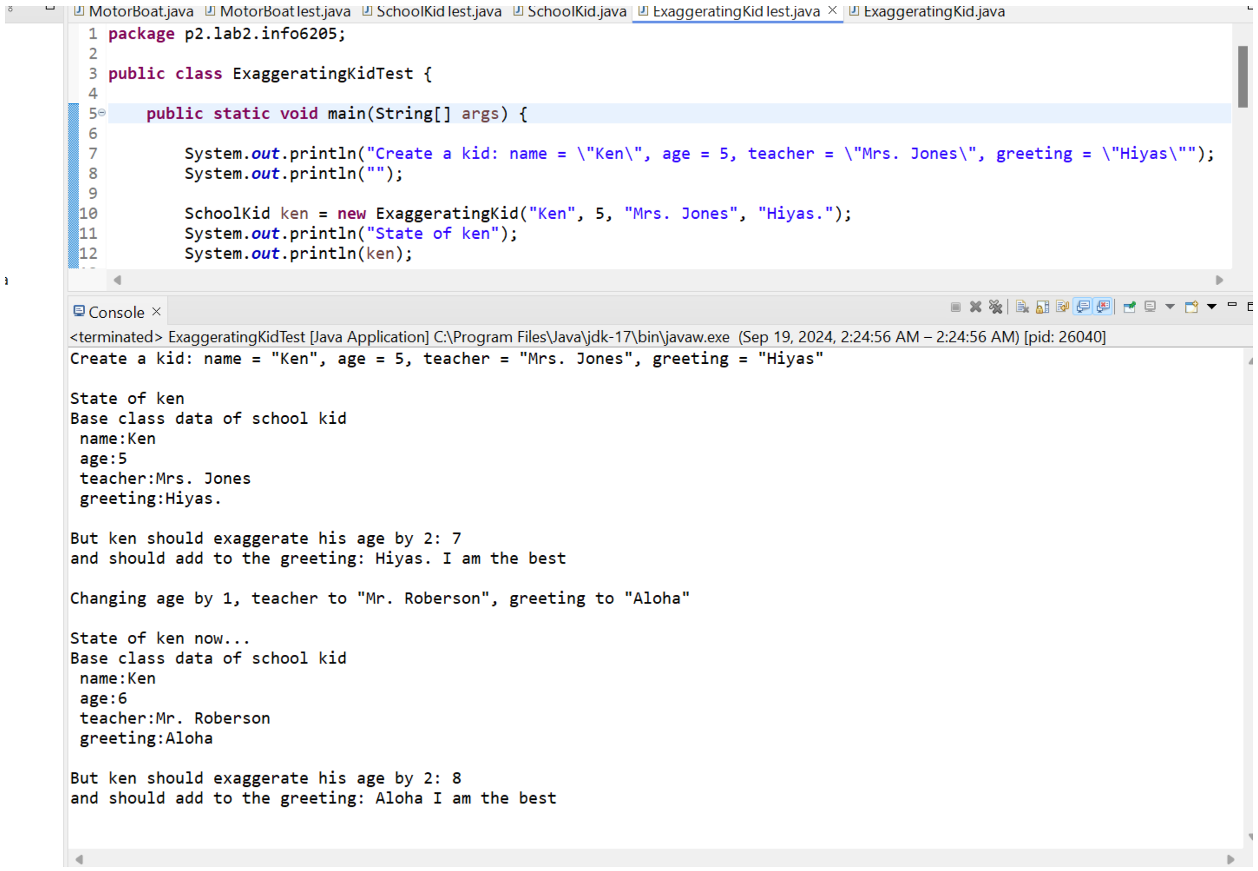
Analysis:
I created the child class ExaggeratingKid which extends from SchoolKid. All the SchoolKid properties are available in ExaggeratingKid class. After overriding the getAge() method, I provided implementation in increase the age value by 2. Similarly, for getGreeting() method I provided the implementation by concatenating "I am the best" at the end.
No difficulties were faced, as this was a smooth code. The inheritance concept was used here which gave us a good warmup for the java concepts.
As we are changing functionality of getAge() and getGreeting() we need to override these methods from the parent class. Solution looks good

Source code:

```
// ExaggeratingKid.java
package p2.lab2.info6205;

public class ExaggeratingKid extends SchoolKid {

    public ExaggeratingKid() {

    }

    public ExaggeratingKid(String name, int age, String teacher, String greeting) {
            super(name, age, teacher, greeting);
    }

    @Override
    public int getAge() {
            return super.getAge() + 2;
    }

    @Override
    public String getGreeting() {
            return super.getGreeting() + " I am the best";
    }
}
```

```
}


// ExaggeratingKidTest.java
package p2.lab2.info6205;

public class ExaggeratingKidTest {

    public static void main(String[] args) {

            System.out.println("Create a kid: name = \"Ken\", age = 5, teacher = \"Mrs.
            Jones\", greeting = \"Hiyas\"");
            System.out.println("");

            SchoolKid ken = new ExaggeratingKid("Ken", 5, "Mrs. Jones", "Hiyas.");
            System.out.println("State of ken");
            System.out.println(ken);

            System.out.println("");

            System.out.println("But ken should exaggerate his age by 2: " +
            ken.getAge());
            System.out.println("and should add to the greeting: " + ken.getGreeting());

            System.out.println("");

            System.out.println("Changing age by 1, teacher to \"Mr. Roberson\", greeting
            to \"Aloha\"");

            ken.haveBirthday();
            ken.changeTeacher("Mr. Roberson");
            ken.changeGreeting("Aloha");

            System.out.println("");

            System.out.println("State of ken now...");
            System.out.println(ken);

            System.out.println("");

            System.out.println("But ken should exaggerate his age by 2: " +
            ken.getAge());
            System.out.println("and should add to the greeting: " + ken.getGreeting());

      }

}
```

Screenshots/Output:



```java
package p2.lab2.info6205;

public class ExaggeratingKidTest {

    public static void main(String[] args) {

        System.out.println("Create a kid: name = \"Ken\", age = 5, teacher = \"Mrs. Jones\", greeting = \"Hiyas\"");
        System.out.println("");

        SchoolKid ken = new ExaggeratingKid("Ken", 5, "Mrs. Jones", "Hiyas.");
        System.out.println("State of ken");
        System.out.println(ken);
```

```
<terminated> ExaggeratingKidTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Sep 19, 2024, 2:24:56 AM – 2:24:56 AM) [pid: 26040]
Create a kid: name = "Ken", age = 5, teacher = "Mrs. Jones", greeting = "Hiyas"

State of ken
Base class data of school kid
 name:Ken
 age:5
 teacher:Mrs. Jones
 greeting:Hiyas.

But ken should exaggerate his age by 2: 7
and should add to the greeting: Hiyas. I am the best

Changing age by 1, teacher to "Mr. Roberson", greeting to "Aloha"

State of ken now...
Base class data of school kid
 name:Ken
 age:6
 teacher:Mr. Roberson
 greeting:Aloha

But ken should exaggerate his age by 2: 8
and should add to the greeting: Aloha I am the best
```