

Lab6

Yiling Hu | 001537980

## I. Classes, Predicates, Methods

1. What is the data type of the variable `ourDogs`?

It is a `List of Dogs`.

2. What is the `size()` of `ourDogs`, and what are the elements it contains?

2 elements(`Dog("Louise", false, 5)` and `Dog("Coco", false, 7)`)

3. Assume we change our predicate in the above example to:

```
Predicate<Dog> dp = (d) -> d.isMale() && d.getAge() > 5;
```

Now, what is the `size()` of `ourDogs`, and what are the elements it contains?

1 element(`Dog("Fido", true, 10)`)

## II. Implement Code

```
/**
 * This class represents the transmission box of an automatic car. The
 * transmission box
 * has a current speed, a current gear and 4 speed thresholds for the 5 gears
 * in order.
 */

public class TransmissionBox {

    private int currentSpeed; // current speed of the transmission box
    private int currentGear; // current gear

    // Speed thresholds representing going from 1 to 2, or 2 to 3 (or back),
    etc.
    private final int thresholdOne;
    private final int thresholdTwo;
    private final int thresholdThree;
    private final int thresholdFour;

    /**
     * Construct a TransmissionBox object and initializes it to the current
     * speed and 4
     * speed thresholds.
     * @param currentSpeed the current speed
     * @param thresholdOne the first threshold, representing going from 1 to
     2 or back.
     * @param thresholdTwo the second threshold, representing going from 2 to
     3 or back.
     * @param thresholdThree the third threshold, representing going from 3
     to 4 or back.
     * @param thresholdFour the fourth threshold, representing going from 4
```

```

to 5 or back.
    */
    public TransmissionBox(int currentSpeed, int thresholdOne,
                           int thresholdTwo, int thresholdThree,
                           int thresholdFour) throws IllegalArgumentException
    {
        if (currentSpeed < 0 || currentSpeed > 300) throw new
IllegalArgumentException("The current speed is invalid");
        this.currentSpeed = currentSpeed;
        if (thresholdOne < 0) throw new IllegalArgumentException("The
threshold one setting is invalid");
        this.thresholdOne = thresholdOne;
        if (thresholdTwo <= thresholdOne) throw new
IllegalArgumentException("The threshold two setting is invalid");
        this.thresholdTwo = thresholdTwo;
        if (thresholdThree <= thresholdTwo) throw new
IllegalArgumentException("The threshold three setting is invalid");
        this.thresholdThree = thresholdThree;
        if (thresholdFour <= thresholdThree) throw new
IllegalArgumentException("The threshold four setting is invalid");
        this.thresholdFour = thresholdFour;
        if (currentSpeed >= thresholdFour) currentGear = 5;
        else if (currentSpeed >= thresholdThree) currentGear = 4;
        else if (currentSpeed >= thresholdTwo) currentGear = 3;
        else if (currentSpeed >= thresholdOne) currentGear = 2;
        else currentGear = 1;
    }

    /**
     * Returns a TransmissionBox object with speed increased by 2 and the
appropriate gear.
     * @return a TransmissionBox object with speed increased by 2 and the
appropriate gear.
     */
    public TransmissionBox speedIncrease() {
        int newSpeed = this.currentSpeed * 2;
        TransmissionBox newTransmissionBox = new TransmissionBox(newSpeed,
thresholdOne,thresholdTwo, thresholdThree, thresholdFour);
        if (newSpeed >= thresholdFour) newTransmissionBox.currentGear = 5;
        else if (newSpeed >= thresholdThree) newTransmissionBox.currentGear =
4;
        else if (newSpeed >= thresholdTwo) newTransmissionBox.currentGear =
3;
        else if (newSpeed >= thresholdOne) newTransmissionBox.currentGear =
2;
        else newTransmissionBox.currentGear = 1;
        return newTransmissionBox;
    }

    /**
     * Returns a TransmissionBox object with speed decreased by 2 and the
appropriate gear.
     * @return a TransmissionBox object with speed decreased by 2 and the
appropriate gear.
     */
    public TransmissionBox speedDecrease() {
        int newSpeed = this.currentSpeed / 2;

```

```

        TransmissionBox newTransmissionBox = new TransmissionBox(newSpeed,
thresholdOne,thresholdTwo, thresholdThree, thresholdFour);
        if (newSpeed >= thresholdFour) newTransmissionBox.currentGear = 4;
        else if (newSpeed >= thresholdThree) newTransmissionBox.currentGear =
3;
        else if (newSpeed >= thresholdTwo) newTransmissionBox.currentGear =
2;
        else if (newSpeed >= thresholdOne) newTransmissionBox.currentGear =
1;
        else newTransmissionBox.currentGear = 0;
        return newTransmissionBox;
    }

    /**
     * Returns the current speed of a TransmissionBox.
     * @return currentSpeed.
     */
    public int getSpeed() { return this.currentSpeed;}

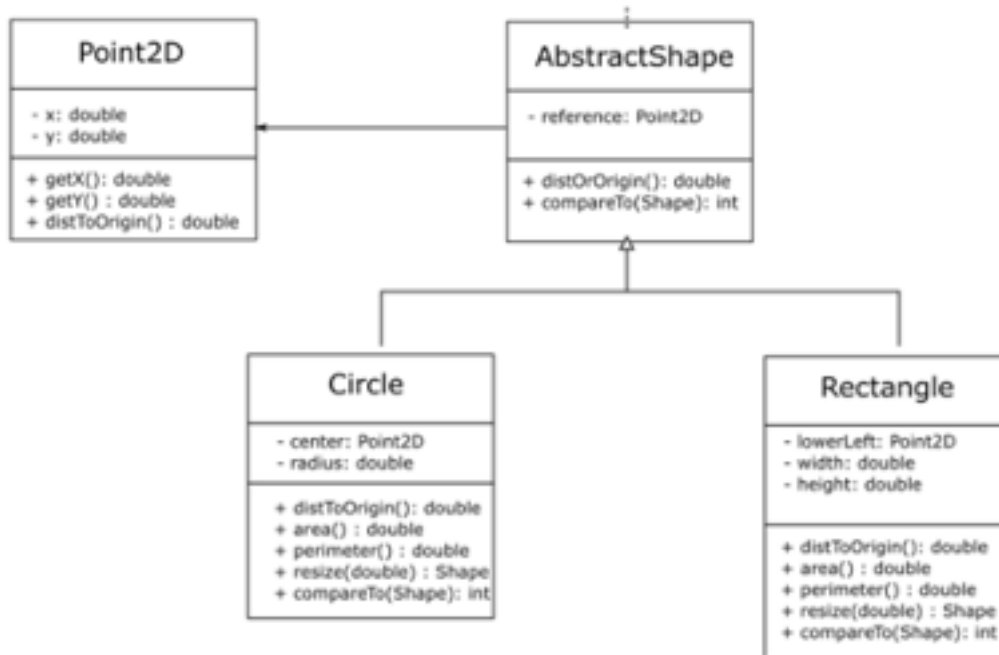
    /**
     * Returns the current gear of a TransmissionBox.
     * @return currentGear.
     */
    public int getCurrentGear() { return this.currentGear;}

    /**
     * Returns a string with the current speed and current gear of the
TransmissionBox.
     * @return str.
     */

    @Override
    public String toString() {
        return "The current speed is " + currentSpeed + " and the current
gear is " + currentGear;
    }
}

```

### III. Conceptual Design



1.What relationship does AbstractShape have with Point2D?

Association relationship, AbstractShape has a Point2D

Using a UML class diagram create a design for the following (you can defer declaring the types for any instance variables as long as you represent the concepts appropriately):

A Person may own zero or more Residences. Each residence has some mortgage amount, and each residence is able to automatically have its taxes assessed. A residence is at a single Location, which has an address and insurance rating. Finally, there are two specific types of residences we are concerned with (for now): Houses and Condominiums.

