Activity: Dynamic or Static?

Use the code in this attachment to follow along with the slides and answer the questoins.

UML Diagram

First, draw the UML diagram for Sinkable, Boat, MotorBoat, and SailBoat in the space below.

You can **leave the classes blank (**omit attributes & methods**)**, but make sure to **draw the relationships** between classes and if they are an interface/abstract class!

Sinkable.java

```java
public interface Sinkable {
    void sink();
}
```

Pos.java **(**don't need to include in UML**)**

```java
public class Pos {
    int x;
    int y;

    public Pos(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public String toString() {
        return "x: " + x + " y: " + x;
    }
}
```

Boat.java

```
public abstract class Boat implements Sinkable {

    private static int count = 0;
    private String name;
    private Pos position;
    private boolean isAboveWater;


    public Boat(String bName) {
        this.name = bName;
        position = new Pos(0, 0);
        isAboveWater = true;
        count = count + 1;
    }

    public String getName() {
        return name;
    }

    public Pos getPos() {
        return position;
    }

    public int getCount() {
        return count;
    }

    public void setPos(Pos newPos) {
        if (isAboveWater) {
            position = newPos;
        }
    }

    public abstract Pos move();

    public void sink() {
        isAboveWater = false;
    }

    public String toString() {
        return String.format("Boat: %s %s Count: %d", name, position.toString(), count);
    }
}
```

MotorBoat.java

```
public class MotorBoat extends Boat {

    int speed;

    public MotorBoat(String mbName, int speed) {
        super(mbName);
        this.speed = speed;
    }

    public Pos move() {
        int newX = super.getPos().getX() + speed;
        int newY = super.getPos().getY() + speed;
        Pos newPos = new Pos(newX, newY);
        super.setPos(newPos);
        return super.getPos();
    }

}
```

SailBoat.java

```java
public class SailBoat extends Boat {

    public static final int WINDSPEED = 10;

    private int sailNum;

    public SailBoat(String sName, int sailNum) {
        super(sName);
        this.sailNum = sailNum;
    }

    public Pos move() {
        int newX = super.getPos().getX() + WINDSPEED;
        int newY = super.getPos().getY() + WINDSPEED;
        Pos newPos = new Pos(newX, newY);
        super.setPos(newPos);
        return super.getPos();
    }

    public Pos sail() {
        Pos newPos = super.getPos();
        for (int i = 0; i < sailNum; i++) {
            newPos = move();
        }
        return newPos;
    }

    public String toString() {
        return String.format("Sailboat: %s %d", super.getName(), sailNum);
    }
}
```