

Lab 9 Sample Test Case

Question 1

```
public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    try {
        c.openOneDoor(8);
    } catch (BadDoorException ex) {
        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
    try {
        c.openOneDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 1);
}

public static void testDoor() {
    Door d = new Door();
    System.out.println(d.isOpen() == false);
    d.close();
    System.out.println(d.isOpen() == false);
    d.open();
    System.out.println(d.isOpen() == true);
}
```

Question 2

```
public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.changeAllDoors();
}
```

```

System.out.println(c.countOpenDoors() == 7);
c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 0);
try {
    c.openOneDoor(8);
} catch (BadDoorException ex) {
    System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
}
try {
    c.openOneDoor(3);
} catch (BadDoorException ex) {
    System.out.println("BUG! This must never happen!");
}
System.out.println(c.countOpenDoors() == 1);
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 6);
c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 1);
c.listDoors();
}

public static void testDoor() {
    Door d = new Door();
    System.out.println(d.isOpen() == false);
    d.close();
    System.out.println(d.isOpen() == false);
    d.open();
    System.out.println(d.isOpen() == true);
}

```

Question 3

```

public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    try {
        c.openOneDoor(8);
    } catch (BadDoorException ex) {
        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
    try {
        c.openOneDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
}

```

```

    }
    System.out.println(c.countOpenDoors() == 1);

    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 6);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 1);
    c.listDoors();
    try {
        c.replaceDoor(8);
    } catch (BadDoorException ex) {

        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }

    try {
        c.replaceDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 0);

    c.listDoors();
}
}

public static void testDoor() {
    Door d = new Door();
    System.out.println(d.isOpen() == false);
    d.close();
    System.out.println(d.isOpen() == false);
    d.open();
    System.out.println(d.isOpen() == true);
}

```

Question 4

```

public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    try {
        c.openOneDoor(8);
    } catch (BadDoorException ex) {
        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
}

```

```

    }
    try {
        c.openOneDoor(3);
    } catch(BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 1);

    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 6);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 1);
    c.listDoors();
    try {
        c.replaceDoor(8);
    } catch(BadDoorException ex) {

        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }

    try {
        c.replaceDoor(3);
    } catch(BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 0);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.replaceAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.listDoors();
}

public static void testDoor() {
    Door d = new Door();
    System.out.println(d.isOpen() == false);
    d.close();
    System.out.println(d.isOpen() == false);
    d.open();
    System.out.println(d.isOpen() == true);
}

```

Question 5

```

public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch(BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch(BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
}

```

```

c.changeAllDoors();
System.out.println(c.countOpenDoors() == 7);
c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 0);
try {
    c.openOneDoor(8);
} catch (BadDoorException ex) {
    System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
}
try {
    c.openOneDoor(3);
} catch (BadDoorException ex) {
    System.out.println("BUG! This must never happen!");
}
System.out.println(c.countOpenDoors() == 1);

c.changeAllDoors();
System.out.println(c.countOpenDoors() == 6);
c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 1);
c.listDoors();
try {
    c.replaceDoor(8);
} catch (BadDoorException ex) {

    System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
}

try {
    c.replaceDoor(3);
} catch (BadDoorException ex) {
    System.out.println("BUG! This must never happen!");
}
System.out.println(c.countOpenDoors() == 0);

c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 7);
c.listDoors();
c.replaceAllDoors();
System.out.println(c.countOpenDoors() == 0);
c.listDoors();
c.changeAllDoors();
System.out.println(c.countOpenDoors() == 7);
c.listDoors();
c.replaceManyDoors(4);
System.out.println(c.countOpenDoors() == 3);
c.listDoors();
try {
    c.replaceManyDoors(20);
} catch (ArrayIndexOutOfBoundsException ex) {
    System.out.println(ex.getMessage().equals("Index 7 out of bounds for length
7"));
}

System.out.println(c.countOpenDoors() == 0);
c.listDoors();
}

public static void testDoor() {
    Door d = new Door();

```

```

        System.out.println(d.isOpen() == false);
        d.close();
        System.out.println(d.isOpen() == false);
        d.open();
        System.out.println(d.isOpen() == true);
    }
}

```

Question 6

```

public static void testCar() {
    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    try {
        c.openOneDoor(8);
    } catch (BadDoorException ex) {
        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
    try {
        c.openOneDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 1);

    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 6);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 1);
    c.listDoors();
    try {
        c.replaceDoor(8);
    } catch (BadDoorException ex) {

        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }

    try {
        c.replaceDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 0);

    c.listDoors();
}

```

```

        c.changeAllDoors();
        System.out.println(c.countOpenDoors() == 7);
        c.listDoors();
        c.replaceAllDoors();
        System.out.println(c.countOpenDoors() == 0);
        c.listDoors();
        c.changeAllDoors();
        System.out.println(c.countOpenDoors() == 7);
        c.listDoors();
        c.replaceManyDoors(4);
        System.out.println(c.countOpenDoors() == 3);
        c.listDoors();
        try {
            c.replaceManyDoors(20);
        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.println(ex.getMessage().equals("Index 7 out of bounds for length
7"));
        }

        System.out.println(c.countOpenDoors() == 0);
        c.listDoors();

        c.expandCar();
        c.changeAllDoors();
        System.out.println(c.countOpenDoors() == 9);
        c.listDoors();
    }
    public static void testDoor() {
        Door d = new Door();
        System.out.println(d.isOpen() == false);
        d.close();
        System.out.println(d.isOpen() == false);
        d.open();
        System.out.println(d.isOpen() == true);
    }
}

```

Question 7

```

public static void testCar() {

    try {
        Car brokenCar = new Car("Broken", 0);
    } catch (BadCarException ex) {
        System.out.println(ex.getMessage() == "A car must have at least one door!");
    }
    Car c = null;
    try {
        c = new Car("Biggy", 7);
    } catch (BadCarException ex) {
        System.out.println("BUG! This must never happen!");
    }
    c.listDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    try {

```

```

        c.openOneDoor(8);
    } catch (BadDoorException ex) {
        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
    try {
        c.openOneDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 1);

    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 6);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 1);
    c.listDoors();
    try {
        c.replaceDoor(8);
    } catch (BadDoorException ex) {

        System.out.println(ex.getMessage().equals("Door 8 does not exist!"));
    }
    try {
        c.replaceDoor(3);
    } catch (BadDoorException ex) {
        System.out.println("BUG! This must never happen!");
    }
    System.out.println(c.countOpenDoors() == 0);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.replaceAllDoors();
    System.out.println(c.countOpenDoors() == 0);
    c.listDoors();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 7);
    c.listDoors();
    c.replaceManyDoors(4);
    System.out.println(c.countOpenDoors() == 3);
    c.listDoors();
    try {
        c.replaceManyDoors(20);
    } catch (IndexOutOfBoundsException ex) {
        System.out.println(ex.getMessage().equals("Index 7 out of bounds for length
7"));
    }
    System.out.println(c.countOpenDoors() == 0);
    c.listDoors();
    c.expandCar();
    c.changeAllDoors();
    System.out.println(c.countOpenDoors() == 9);
    c.listDoors();
}

public static void testDoor() {
    Door d = new Door();
    System.out.println(d.isOpen() == false);
    d.close();
    System.out.println(d.isOpen() == false);
    d.open();
    System.out.println(d.isOpen() == true);
}

```

