

## Practice Exercise

### ArrayList & Modeling class

Design and Implement a Java program as follows:

- 1) Model Candle object to include at minimum the following:
  - Member variables (attributes): color (e.g. “Blue”), height (e.g. 1 inch), diameter (e.g. 1.5 inch), indication if lit (e.g. true or false)
  - Functionality: change height; light candle, extinguish candle
  - New candle on creation is unlit
- 2) Create HandleCandles class with main method which stores a list of candles.
- 3) In main method display a menu where the user can:
  - Add new candle instance
  - Remove a candle instance (based on height and color)
  - Light or extinguish candle instance (based on color and height)
  - Display candles

**Hints:** Use Java’s builtin ArrayList or Vector to store a list of Candle instances

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

<https://docs.oracle.com/javase/8/docs/api/java/util/Vector.html>

### SAMPLE RUN 1

```
MENU
1: Add candle
2: Remove candle
3: Light candle
4: Extinguish candle
5: Display candles
9: Exit program

Enter your selection : 1
What is the candle's height? 2
What is the candle's diameter? 1.3
What is the candle's color? blue

The following candle was created: Candle [color=blue, height=2, diameter=1.3,
isLit=false]

MENU
1: Add candle
2: Remove candle
3: Light candle
4: Extinguish candle
5: Display candles
9: Exit program

Enter your selection : 9

Thank you for using the program. Goodbye!
```

## SAMPLE RUN 2

```
MENU
1: Add candle
2: Remove candle
3: Light candle
4: Extinguish candle
5: Display candles
9: Exit program

Enter your selection : 1
What is the candle's height? 2
What is the candle's diameter? 1.2
What is the candle's color? blue

The following candle was created: Candle [color=blue, height=2, diameter=1.2,
isLit=false]
```

```
MENU
1: Add candle
2: Remove candle
3: Light candle
4: Extinguish candle
5: Display candles
9: Exit program

Enter your selection : 1
What is the candle's height? 3
What is the candle's diameter? 1.4
What is the candle's color? red

The following candle was created: Candle [color=red, height=3, diameter=1.4,
isLit=false]
```

```
MENU
1: Add candle
2: Remove candle
3: Light candle
4: Extinguish candle
5: Display candles
9: Exit program

Enter your selection : 5

Candle [color=blue, height=2, diameter=1.2, isLit=false]
Candle [color=red, height=3, diameter=1.4, isLit=false]
```

## SAMPLE RUN 3

[Adding 2 candles as above]

```
MENU
1: Add candle
2: Remove candle
```

3: Light candle  
4: Extinguish candle  
5: Display candles  
9: Exit program

Enter your selection : 3

What is the height of the candle you want to light/extinguish? 3

What is color of the candle you want to light/extinguish? red

Lit/extinguished candle: Candle [color=red, height=3, diameter=1.4, isLit=true]

## **SAMPLE RUN 4**

### **[Adding 2 candles as above]**

MENU

1: Add candle  
2: Remove candle  
3: Light candle  
4: Extinguish candle  
5: Display candles  
9: Exit program

Enter your selection : 2

What is the height of the candle you want to remove? 2

What is color of the candle you want to remove? blue

Removed the following candle: Candle [color=blue, height=2, diameter=1.2, isLit=false]

MENU

1: Add candle  
2: Remove candle  
3: Light candle  
4: Extinguish candle  
5: Display candles  
9: Exit program

Enter your selection : 5

Candle [color=red, height=3, diameter=1.4, isLit=false]

## Sample Solution:

```
public class Candle {

    // attributes
    public String color;
    public int height;
    public double diameter;
    public boolean isLit;

    // constructor
    public Candle(String color, int height, double diameter) {

        // Can do validation that height and diameter are not less than 0
        if (height <= 0 || diameter <= 0) {
            System.out.println("Height and diameter must be greater than 0. No values have
been set");
            return; // this should be an exception to be covered later in the course
        }

        // set values
        this.color = color;
        this.height = height;
        this.diameter = diameter;

        // candle state must be unlit
        isLit = false;
    }

    // method to light candle
    public void lightCandle() {

        if (isLit) // validation
            System.out.println("Candle is already lit");
        else
            isLit = true;
    }

    // method to extinguish candle
    public void extinguishCandle() {

        if (!isLit) // validation
            System.out.println("Candle is already extinguished");
        else
            isLit = false;
    }

    // method to change height (e.g. as it burns may need to update height)
    public void updateHeight(int height) {

        // validate
        if (height > this.height || height <= 0)
            System.out.println("New candle height must be less than current height but
greater than 0");
        else
            this.height = height;
    }

    public String toString() {
        return "Candle [color=" + color + ", height=" + height + ", diameter=" + diameter + ",
isLit=" + isLit + "]\n";
    }
}
```

```

import java.util.ArrayList;
import java.util.Scanner;

public class HandleCandles {

    ArrayList<Candle> list; // attribute to store a list of Candle objects

    // constructor
    public HandleCandles() {

        list = new ArrayList<Candle>(); // creates new instance of an empty list
    }

    // method to display the menu
    public void displayMenu() {

        System.out.println("\n    MENU");
        System.out.println("1: Add candle ");
        System.out.println("2: Remove candle ");
        System.out.println("3: Light candle ");
        System.out.println("4: Extinguish candle ");
        System.out.println("5: Display candles ");
        System.out.println("9: Exit program");

    }

    // method to handle user's selection
    public void processChoice(int c) {

        switch (c) {

            case 1 :      addCandle();
                          break;
            case 2 :      removeCandle();
                          break;
            case 3 :      lightCandle(true);
                          break;
            case 4 :      lightCandle(false);
                          break;
            case 5 :      displayCandles();
                          break;
            case 9:      System.out.println("\nThank you for using the program. Goodbye!");
                          break;
            default:      System.out.println("Invalid choice");

        }

    }

    // method to display all candle information in the list
    private void displayCandles() {

        if (list.size() == 0)
            System.out.println("\nThere are no candles to display");
        else {
            System.out.println(); // empty line before candle data
            for (int i=0; i < list.size(); i++) { // loop for all candles in the list
                Candle c = list.get(i); // get candle instance from the list for each
index value
                System.out.println(c.toString()); // print candle data to console
            }
        }

    }

    // when light=true, lit the candle and if light=false, extinguish candle
    // TODO: could also first check if already lit or extinguished and give message
    private void lightCandle(boolean light) {

```

```

Scanner stdin = new Scanner(System.in);

// prompt user for data to find the candle to remove
System.out.print("What is the height of the candle you want to light/extinguish? ");
int height = stdin.nextInt();
System.out.print("What is color of the candle you want to light/extinguish? ");
String color = stdin.next();

// look for the candle that matches above
int size = list.size(); // number of candles in the list
System.out.println(); // empty line before candle data
for (int i=0; i < size; i++) { // loop for all candles in the list
    Candle c = list.get(i); // get candle instance from the list for each index
    if (c.height == height && c.color.equalsIgnoreCase(color)) {
        if (light)
            c.isLit = true; // light candle
        else
            c.isLit = false; // extinguish candle
        list.set(i, c); // update in list
        System.out.println("\nLit/extinguished candle: " + c.toString()); //
    }

    }

// if here than did not find the candle
System.out.println("\nThere is no candle with this height and color");

}

// method to remove the first candle that matches user's criteria
private void removeCandle() {

    Scanner stdin = new Scanner(System.in);

    // prompt user for data to find the candle to remove
    System.out.print("What is the height of the candle you want to remove? ");
    int height = stdin.nextInt();
    System.out.print("What is color of the candle you want to remove? ");
    String color = stdin.next();

    // look for the candle that matches above
    int size = list.size(); // number of candles in the list
    System.out.println(); // empty line before candle data
    for (int i=0; i < size; i++) { // loop for all candles in the list
        Candle c = list.get(i); // get candle instance from the list for each index
        if (c.height == height && c.color.equalsIgnoreCase(color)) {
            list.remove(i); // remove the candle
            System.out.println("\nRemoved the following candle: " + c.toString());
        }

        }

    // if did not return from for-loop that mean sit did not find the candle
    System.out.println("\nThere is no candle with this height and color");

}

private void addCandle() {

    Scanner stdin = new Scanner(System.in);

    // prompt user for all candle data to be set on creation
    System.out.print("What is the candle's height? ");
    int height = stdin.nextInt();

```

```

        System.out.print("What is the candle's diameter? ");
        double diameter = stdin.nextDouble();
        System.out.print("What is the candle's color? ");
        String color = stdin.next();

        // candle must start as unlit so not prompting

        // create Candle instance
        Candle c = new Candle(color, height, diameter);

        // tell user what was created
        System.out.println("\nThe following candle was created: " + c.toString());

        // add candle instance to list
        list.add(c);
    }

    public static void main(String[] args) {

        HandleCandles handler = new HandleCandles(); // new instance of driver class

        Scanner stdin = new Scanner(System.in);
        int selection = 0;

        do {

            handler.displayMenu();

            System.out.print("\nEnter your selection : ");
            selection = stdin.nextInt();

            handler.processChoice(selection);

        } while (selection != 9);

        stdin.close();
    }
}

```