Jinan University

Java Programming Lab Report

Major: Computer Science & Technology
School: International School

Student name : ____
(p.s. your name on JNU academic system)

Student number : ____

Date of Submission (mm-dd-yyyy): __________________________

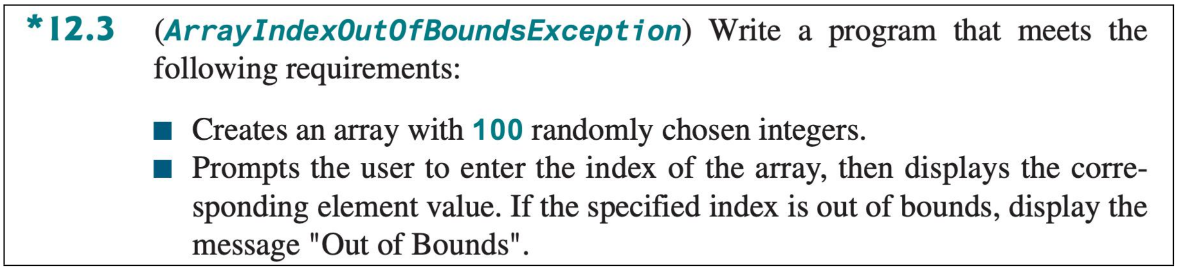
Instructor: Yuxia Sun

Table of Content

LAB 8 DATE: 6/6/2023

Student Name:______________ Student ID: _______________

Problem 1. (12.3)



***12.3** (*ArrayIndexOutOfBoundsException*) Write a program that meets the following requirements:

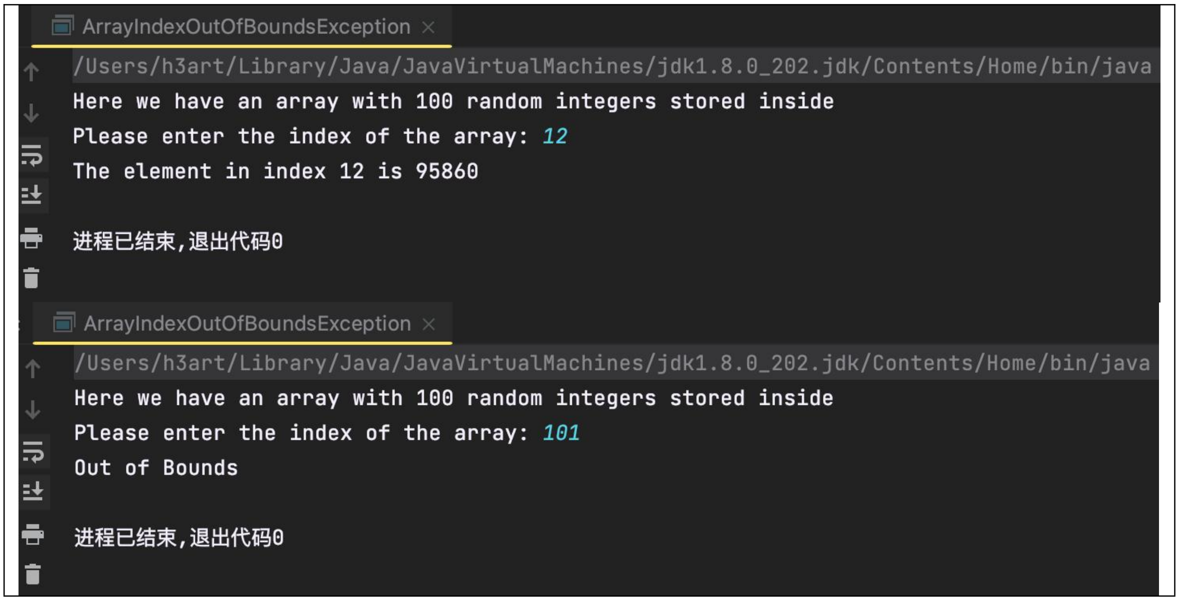
- Creates an array with **100** randomly chosen integers.
- Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message "Out of Bounds".

**** Source Code / Solution :***

```java
import java.util.Scanner;

public class ArrayIndexOutOfBoundsException {

    private final static int MAXLEN = 100;
    private static final int[] array = new int[MAXLEN];

    private static void randomIntGenerator() {
        for (int i = 0; i < MAXLEN; i++) {
            array[i] = (int) (100000 * Math.random());
        }
    }

    public static void main(String[] args) {
        try (Scanner input = new Scanner(System.in)) {
            randomIntGenerator();
            System.out.println("Here we have an array with 100 random integers stored inside");
            System.out.print("Please enter the index of the array: ");
            int index = input.nextInt();
            System.out.println("The element in index " + index + " is " + array[index]);
        } catch (java.lang.ArrayIndexOutOfBoundsException e) {
            System.out.println("Out of Bounds");
        }
```

```
  }
}
```

**** Output:***


ArrayIndexOutOfBoundsException

```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java
Here we have an array with 100 random integers stored inside
Please enter the index of the array: 12
The element in index 12 is 95860

进程已结束,退出代码0
```

ArrayIndexOutOfBoundsException

```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java
Here we have an array with 100 random integers stored inside
Please enter the index of the array: 101
Out of Bounds

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Redundant try-catch blocks make the code less readable.
Fix: The try-with-resource syntactic sugar added after JDK7 is used to save the step of closing the Scanner.

Problem 2. (12.7)

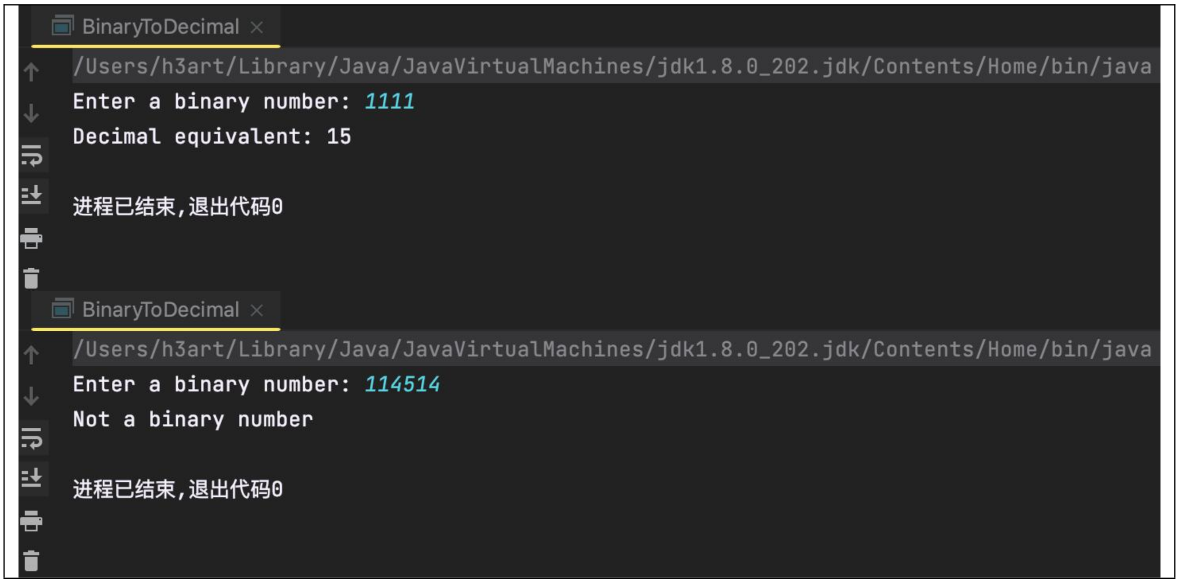
***12.7** (*NumberFormatException*) Write the `bin2Dec(String binaryString)` method to convert a binary string into a decimal number. Implement the `bin2Dec` method to throw a `NumberFormatException` if the string is not a binary string. Write a test program that prompts the user to enter a binary number as a string and displays its decimal equivalent. If the method throws an exception, display “Not a binary number”.

**** Source Code / Solution :***

```java
import java.util.Scanner;

public class BinaryToDecimal {
  public static int bin2Dec(String binaryString) throws NumberFormatException {
    // cool package method
    return Integer.parseInt(binaryString, 2);
  }
```

```java

    public static void main(String[] args) {
        try (Scanner input = new Scanner(System.in)) {
            System.out.print("Enter a binary number: ");
            int decimal = bin2Dec(input.nextLine());
            System.out.println("Decimal equivalent: " + decimal);
        } catch (NumberFormatException e) {
            System.out.println("Not a binary number");
        }
    }
}
```

**** Output:***



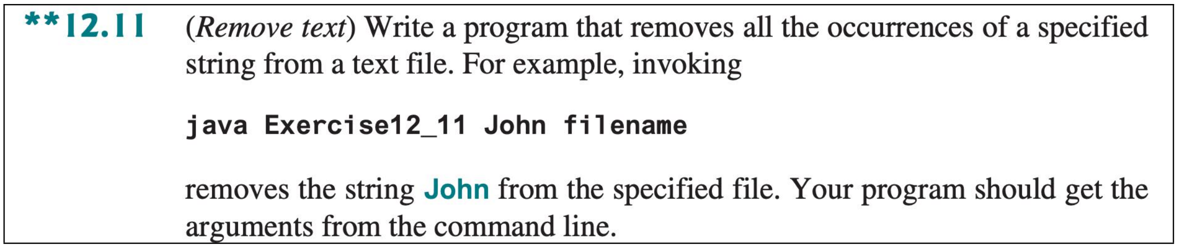
```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java
Enter a binary number: 1111
Decimal equivalent: 15

进程已结束,退出代码0
```

```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java
Enter a binary number: 114514
Not a binary number

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Created a character detection loop body to detect whether the input string meets the parsing conditions of binary numbers, but this is not necessary.
Fix: Delete the loop body, and directly call the parseInt method in the java.lang.Integer package to detect the string and throw NumberFormatException.

<u>Problem 3. (12.11)</u>



****12.11** (*Remove text*) Write a program that removes all the occurrences of a specified string from a text file. For example, invoking

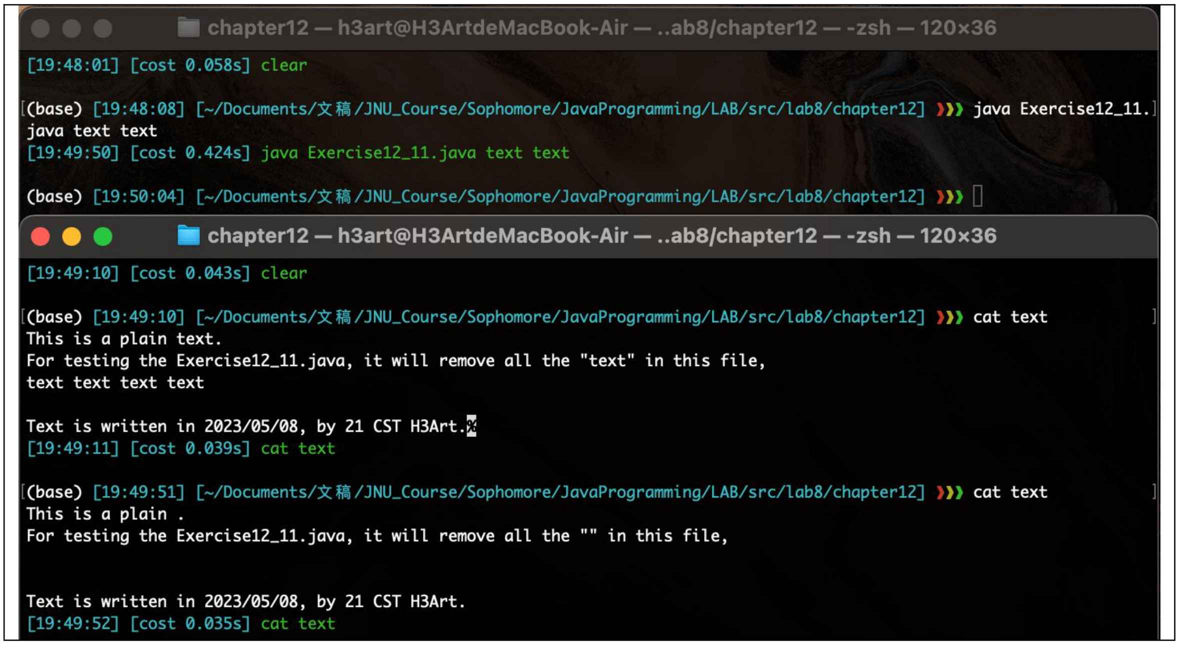
```
java Exercise12_11 John filename
```

removes the string **John** from the specified file. Your program should get the arguments from the command line.

**** Source Code / Solution :***

```
import java.io.*;

public class Exercise12_11 {
    public static void main(String[] args) {
        if (args.length < 2) {
            System.err.println("Usage: java RemoveText <Remove_String> <File_path/File_name.txt>");
            System.exit(1);
        }

        String removeString = args[0];

        for(int i = 1; i < args.length; i++) {
            File targetFile = new File(args[i]);
            try {
                FileReader fileReader = new FileReader(targetFile);
                BufferedReader bufferedReader = new BufferedReader(fileReader);

                // Read all the file, and store data in stringBuilder
                StringBuilder stringBuilder = new StringBuilder();
                String line;

                while ((line = bufferedReader.readLine()) != null) {
                    line = line.replaceAll(removeString, "");
                    stringBuilder.append(line).append("\n");
                }

                bufferedReader.close();
                fileReader.close();

                // Create a fileWriter to write all the data from stringBuilder to file
                FileWriter fileWriter = new FileWriter(targetFile);
                BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
                bufferedWriter.write(stringBuilder.toString());

                bufferedWriter.close();
                fileWriter.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

**** Output:***



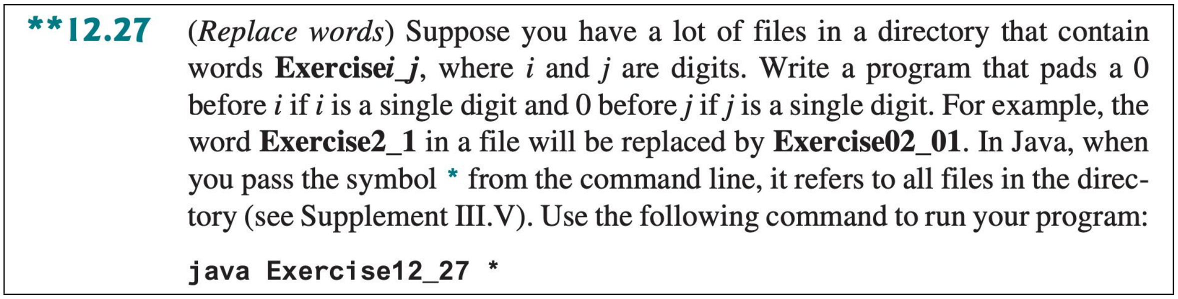
```
chapter12 — h3art@H3ArtdeMacBook-Air — ..ab8/chapter12 — -zsh — 120×36
[19:48:01] [cost 0.058s] clear

(base) [19:48:08] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ java Exercise12_11.
java text text
[19:49:50] [cost 0.424s] java Exercise12_11.java text text

(base) [19:50:04] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯
```

```
chapter12 — h3art@H3ArtdeMacBook-Air — ..ab8/chapter12 — -zsh — 120×36
[19:49:10] [cost 0.043s] clear

(base) [19:49:10] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ cat text
This is a plain text.
For testing the Exercise12_11.java, it will remove all the "text" in this file,
text text text text

Text is written in 2023/05/08, by 21 CST H3Art.%
[19:49:11] [cost 0.039s] cat text

(base) [19:49:51] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ cat text
This is a plain .
For testing the Exercise12_11.java, it will remove all the "" in this file,


Text is written in 2023/05/08, by 21 CST H3Art.
[19:49:52] [cost 0.035s] cat text
```

**** Debugging/Testing:***

Bug1: Forgetting to handle IOException causes the program to fail to compile.
Fix: Add a try-catch statement to handle IOException.

Problem 4. (12.27)(Optional)



****12.27** (*Replace words*) Suppose you have a lot of files in a directory that contain words **Exercise*i*_*j***, where *i* and *j* are digits. Write a program that pads a 0 before *i* if *i* is a single digit and 0 before *j* if *j* is a single digit. For example, the word **Exercise2_1** in a file will be replaced by **Exercise02_01**. In Java, when you pass the symbol * from the command line, it refers to all files in the directory (see Supplement III.V). Use the following command to run your program:

```
java Exercise12_27 *
```

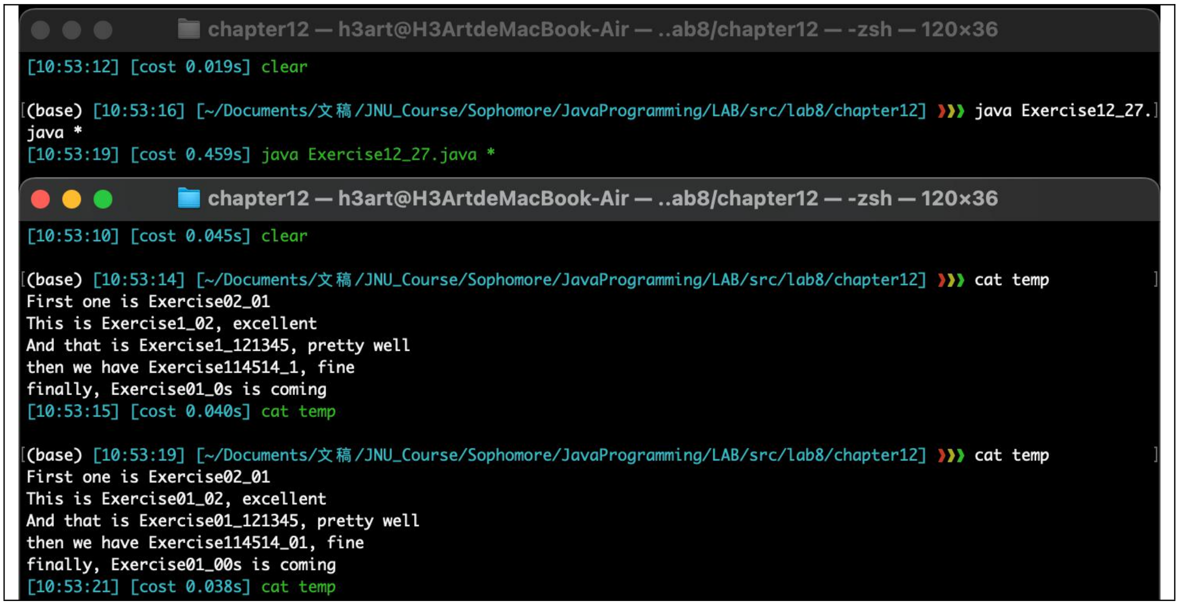
**** Source Code / Solution :***

```
import java.io.*;
import java.util.regex.Pattern;
import java.util.regex.Matcher;


public class Exercise12_27 {
   public static String replaceWord(String line) {
      String matchedPattern = "(Exercise\\d+_\\d+)";

      Pattern regularExpression = Pattern.compile(matchedPattern);

      Matcher matcher = regularExpression.matcher(line);

      while (matcher.find()) {
         String matched = matcher.group(0);

         StringBuilder modified = new StringBuilder(matched);

         // Priorly add 0 to the last one single digit
         if (matched.length() - matched.indexOf('_') == 2) {
            modified.insert(matched.indexOf('_') + 1, '0');
         }

         if (matched.indexOf('_') == 9) {
            modified.insert(8, '0');
         }

         line = line.replaceAll(matched, modified.toString());
      }
```

```
    return line;
}

public static void main(String[] args) {
    if (args.length < 1) {
        System.err.println("Usage: java Exercise12_27 <file_path/file_name.txt>");
        System.exit(1);
    }

    for (String arg : args) {
        File targetFile = new File(arg);
        try {
            FileReader fileReader = new FileReader(targetFile);
            BufferedReader bufferedReader = new BufferedReader(fileReader);

            // Read all the file, and store data in stringBuilder
            StringBuilder stringBuilder = new StringBuilder();
            String line;

            while ((line = bufferedReader.readLine()) != null) {
                line = replaceWord(line);
                stringBuilder.append(line).append("\n");
            }

            bufferedReader.close();
            fileReader.close();

            // Create a fileWriter to write all the data from stringBuilder to file
            FileWriter fileWriter = new FileWriter(targetFile);
            BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
            bufferedWriter.write(stringBuilder.toString());

            bufferedWriter.close();
            fileWriter.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

** Output:*



chapter12 — h3art@H3ArtdeMacBook-Air — ..ab8/chapter12 — -zsh — 120×36

```
[10:53:12] [cost 0.019s] clear

(base) [10:53:16] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ java Exercise12_27.
java *
[10:53:19] [cost 0.459s] java Exercise12_27.java *
```

chapter12 — h3art@H3ArtdeMacBook-Air — ..ab8/chapter12 — -zsh — 120×36

```
[10:53:10] [cost 0.045s] clear

(base) [10:53:14] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ cat temp
First one is Exercise02_01
This is Exercise1_02, excellent
And that is Exercise1_121345, pretty well
then we have Exercise114514_1, fine
finally, Exercise01_0s is coming
[10:53:15] [cost 0.040s] cat temp

(base) [10:53:19] [~/Documents/文稿/JNU_Course/Sophomore/JavaProgramming/LAB/src/lab8/chapter12] ❯❯❯ cat temp
First one is Exercise02_01
This is Exercise01_02, excellent
And that is Exercise01_121345, pretty well
then we have Exercise114514_01, fine
finally, Exercise01_00s is coming
[10:53:21] [cost 0.038s] cat temp
```

** Debugging/Testing:*

Bug1: Inaccurate use of regular expressions leads to matching errors.

Fix: After modifying and verifying the regular expression online test website, write the program so that it can correctly identify the corresponding text fragment.

Bug2: The order of inserting character '0' is improper. After the first character '0' with a smaller index is inserted, the index of subsequent characters will change.

Fix: Insert the character '0' with a larger index into the string first.

Problem 5. (12.33)(Optional)

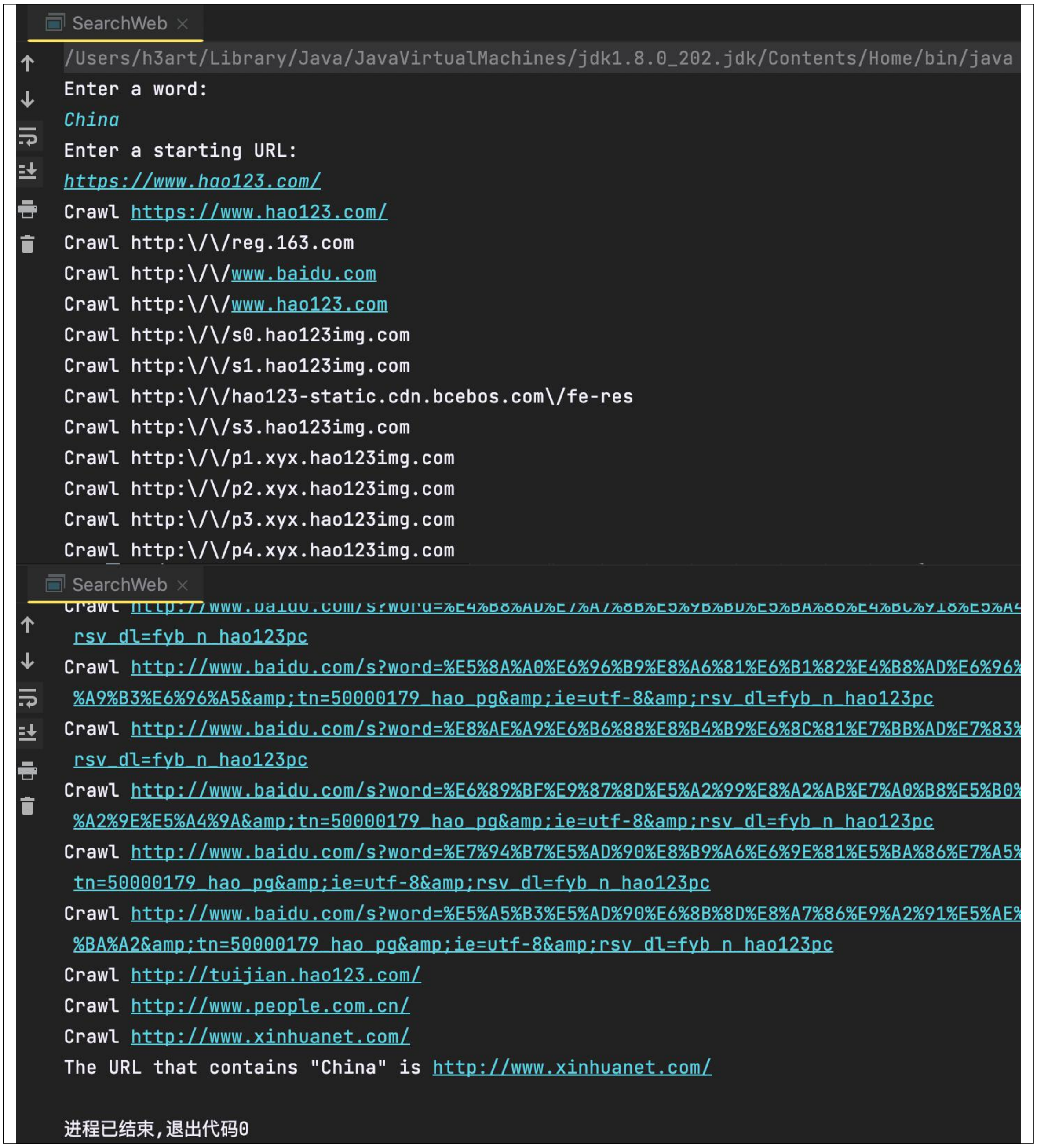
****12.33** (*Search Web*) Modify Listing 12.18 WebCrawler.java to search for the word (e.g., Computer Programming) starting from a URL (e.g., http://cs.armstrong.edu/liang). Your program prompts the user to enter the word and the starting URL and terminates once the word is found. Display the URL for the page that contains the word.

** Source Code / Solution :*

```
import java.util.ArrayList;
import java.util.Objects;
import java.util.Scanner;
import java.net.URL;
```

```
public class SearchWeb {
    public static String searchCorrespondingWord(String startingURL, String word) {
        ArrayList<String> listOfPendingURLs = new ArrayList<>();
        ArrayList<String> listOfTraversedURLs = new ArrayList<>();

        listOfPendingURLs.add(startingURL);

        while (!listOfPendingURLs.isEmpty() && listOfTraversedURLs.size() <= 100) {
            String urlString = listOfPendingURLs.remove(0);

            if (!listOfTraversedURLs.contains(urlString)) {
                listOfTraversedURLs.add(urlString);
                System.out.println("Crawl " + urlString);

                try {
                    URL url = new URL(urlString);
                    Scanner input = new Scanner(url.openStream());
                    String line;
                    int current = 0;

                    while (input.hasNextLine()) {
                        line = input.nextLine();

                        if (line.contains(word)) {
                            return urlString;
                        }

                        current = line.indexOf("http:", current);

                        while (current > 0) {
                            // Set " as endIndex because in html,
                            // an url will be a key-value which store as value="url"
                            int endIndex = line.indexOf("\"", current);

                            // Ensure that a correct URL is found
                            if (endIndex > 0) {
                                // String.substring(begin, end) -> [begin, end)
                                String newUrl = line.substring(current, endIndex);
                                if (!listOfTraversedURLs.contains(newUrl)) {
                                    listOfPendingURLs.add(newUrl);
                                }
                                current = line.indexOf("http:", endIndex);
                            } else {
                                current = -1;
                            }
                        }
                    }
                } catch (Exception e) {
                    System.err.println("Error: " + e.getMessage());
                }
            }

        }

        return "None";
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a word: ");
        String word = input.nextLine();
        System.out.println("Enter a starting URL: ");
        String url = input.nextLine();


        // Traverse the Web from a starting url and search
        String result = searchCorrespondingWord(url, word);

        if (Objects.equals(result, "None")) {
            System.out.println("There are no URL that contains " + word);
        } else {
            System.out.println("The URL that contains \"" + word + "\" is " + result);
        }
    }
}
```

**** Output:***



```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java
Enter a word:
China
Enter a starting URL:
https://www.hao123.com/
Crawl https://www.hao123.com/
Crawl http:\/\/reg.163.com
Crawl http:\/\/www.baidu.com
Crawl http:\/\/www.hao123.com
Crawl http:\/\/s0.hao123img.com
Crawl http:\/\/s1.hao123img.com
Crawl http:\/\/hao123-static.cdn.bcebos.com\/fe-res
Crawl http:\/\/s3.hao123img.com
Crawl http:\/\/p1.xyx.hao123img.com
Crawl http:\/\/p2.xyx.hao123img.com
Crawl http:\/\/p3.xyx.hao123img.com
Crawl http:\/\/p4.xyx.hao123img.com
```

```
Crawl http://www.baidu.com/s?word=%E4%B8%AD%E7%A7%8B%E5%9B%BD%E5%BA%86%E4%BC%91%E5%A4
 rsv_dl=fyb_n_hao123pc
Crawl http://www.baidu.com/s?word=%E5%8A%A0%E6%96%B9%E8%A6%81%E6%B1%82%E4%B8%AD%E6%96%
 %A9%B3%E6%96%A5&amp;tn=50000179_hao_pg&amp;ie=utf-8&amp;rsv_dl=fyb_n_hao123pc
Crawl http://www.baidu.com/s?word=%E8%AE%A9%E6%B6%88%E8%B4%B9%E6%8C%81%E7%BB%AD%E7%83%
 rsv_dl=fyb_n_hao123pc
Crawl http://www.baidu.com/s?word=%E6%89%BF%E9%87%8D%E5%A2%99%E8%A2%AB%E7%A0%B8%E5%B0%
 %A2%9E%E5%A4%9A&amp;tn=50000179_hao_pg&amp;ie=utf-8&amp;rsv_dl=fyb_n_hao123pc
Crawl http://www.baidu.com/s?word=%E7%94%B7%E5%AD%90%E8%B9%A6%E6%9E%81%E5%BA%86%E7%A5%
 tn=50000179_hao_pg&amp;ie=utf-8&amp;rsv_dl=fyb_n_hao123pc
Crawl http://www.baidu.com/s?word=%E5%A5%B3%E5%AD%90%E6%8B%8D%E8%A7%86%E9%A2%91%E5%AE%
 %BA%A2&amp;tn=50000179_hao_pg&amp;ie=utf-8&amp;rsv_dl=fyb_n_hao123pc
Crawl http://tuijian.hao123.com/
Crawl http://www.people.com.cn/
Crawl http://www.xinhuanet.com/
The URL that contains "China" is http://www.xinhuanet.com/

进程已结束,退出代码0
```

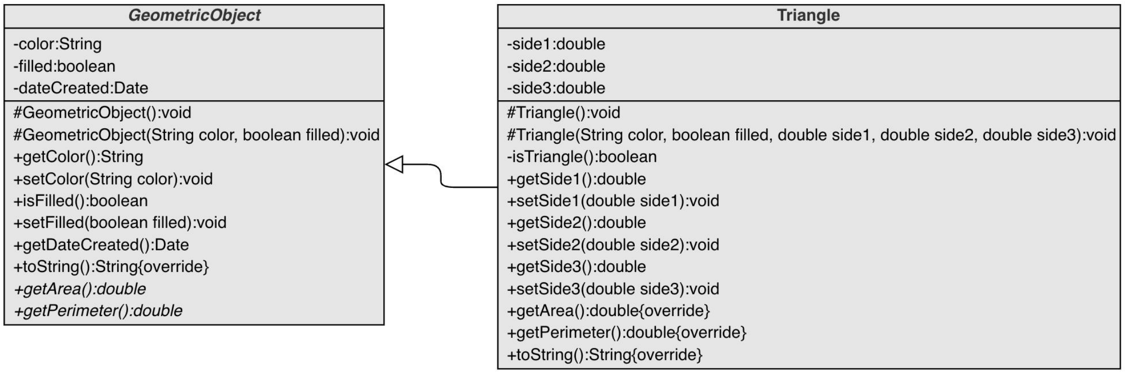
**** Debugging/Testing:***

Bug1: Directly using Scanner's nextLine() method can cause an exception to be thrown when the program cannot read a new line of content, and this situation occurs frequently.
Fix: Before obtaining the content of the next line, make a judgment on the hasNextLine() method before proceeding, to avoid the program throwing a large number of useless exceptions.

Problem 6. (13.1)

**13.1 (*Triangle class*) Design a new Triangle class that extends the abstract GeometricObject class. Draw the UML diagram for the classes Triangle and GeometricObject then implement the Triangle class. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides, and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.

**** Source Code / Solution :***


GeometricObject
-color:String
-filled:boolean
-dateCreated:Date
#GeometricObject():void
#GeometricObject(String color, boolean filled):void
+getColor():String
+setColor(String color):void
+isFilled():boolean
+setFilled(boolean filled):void
+getDateCreated():Date
+toString():String{override}
+getArea():double
+getPerimeter():double
Triangle
-side1:double
-side2:double
-side3:double
#Triangle():void
#Triangle(String color, boolean filled, double side1, double side2, double side3):void
-isTriangle():boolean
+getSide1():double
+setSide1(double side1):void
+getSide2():double
+setSide2(double side2):void
+getSide3():double
+setSide3(double side3):void
+getArea():double{override}
+getPerimeter():double{override}
+toString():String{override}

```java
public abstract class GeometricObject {
  private String color = "white";
  private boolean filled;
  private java.util.Date dateCreated;

  /**
   * Construct a default geometric object
   */
  protected GeometricObject() {
     dateCreated = new java.util.Date();
  }

  /**
   * Construct a geometric object with color and filled value
   */
  protected GeometricObject(String color, boolean filled) {
     dateCreated = new java.util.Date();
     this.color = color;
     this.filled = filled;
  }
```

```
/**
 * Return color
 */
public String getColor() {
    return color;
}


/**
 * Set a new color
 */
public void setColor(String color) {
    this.color = color;
}


/**
 * Return filled. Since filled is boolean,
 * 29 * the getter method is named isFilled
 */
public boolean isFilled() {
    return filled;
}

/**
 * Set a new filled
 */
public void setFilled(boolean filled) {
    this.filled = filled;
}

/**
 * Get dateCreated
 */
public java.util.Date getDateCreated() {
    return dateCreated;

}
```

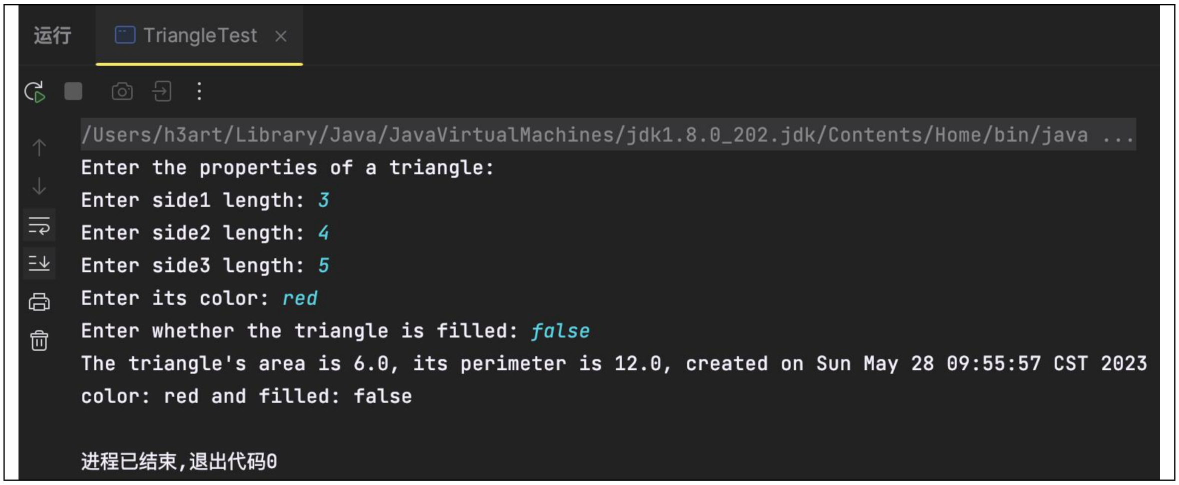
```
  @Override
  public String toString() {
    return "created on " + dateCreated + "\ncolor: " + color +
        " and filled: " + filled;
  }

  /**
   * Abstract method getArea
   */
  public abstract double getArea();

  /**
   * Abstract method getPerimeter
   */
  public abstract double getPerimeter();
}
```

```
public class Triangle extends GeometricObject {
  private double side1;
  private double side2;
  private double side3;

  protected Triangle() {
    super();
    side1 = 1.0;
    side2 = 1.0;
    side3 = 1.0;
  }

  protected Triangle(String color, boolean filled, double side1, double side2, double side3) {
    super(color, filled);
    this.side1 = side1;
    this.side2 = side2;
    this.side3 = side3;
    if (!isTriangle()) {
      throw new IllegalArgumentException("These 3 side cannot form a triangle!");
    }
  }

  private boolean isTriangle() {
        return side1 + side2 > side3 &&
                side1 + side3 > side2 &&
                side2 + side3 > side1;
    }

    public double getSide1() {
        return side1;
    }

    public double getSide2() {
        return side2;
    }

    public double getSide3() {
        return side3;
    }

    public void setSide1(double side1) {
        this.side1 = side1;
    }

    public void setSide2(double side2) {
        this.side2 = side2;
    }

    public void setSide3(double side3) {
        this.side3 = side3;
    }

    @Override
    public double getArea() {
        double factor = this.getPerimeter() / 2;
        return Math.sqrt(factor * (factor - side1) * (factor - side2) * (factor - side3));
    }

    @Override
    public double getPerimeter() {
        return this.side1 + this.side2 + this.side3;
    }
```

```java

  @Override
  public String toString() {
    return "The triangle's area is " + getArea() +
        ", its perimeter is " + getPerimeter() +
        ", " + super.toString();
  }
}
```

```java
import java.util.Scanner;

public class TriangleTest {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    double[] sides = new double[3];
    String color;
    boolean isFilled;

    System.out.println("Enter the properties of a triangle:");
    System.out.print("Enter side1 length: ");
    sides[0] = input.nextDouble();
    System.out.print("Enter side2 length: ");
    sides[1] = input.nextDouble();
    System.out.print("Enter side3 length: ");
    sides[2] = input.nextDouble();
    System.out.print("Enter its color: ");
    color = input.next();
    System.out.print("Enter whether the triangle is filled: ");
    isFilled = input.nextBoolean();
    input.close();

    Triangle triangle = new Triangle(color, isFilled, sides[0], sides[1], sides[2]);

    System.out.println(triangle);
  }
}
```

**** Output:***

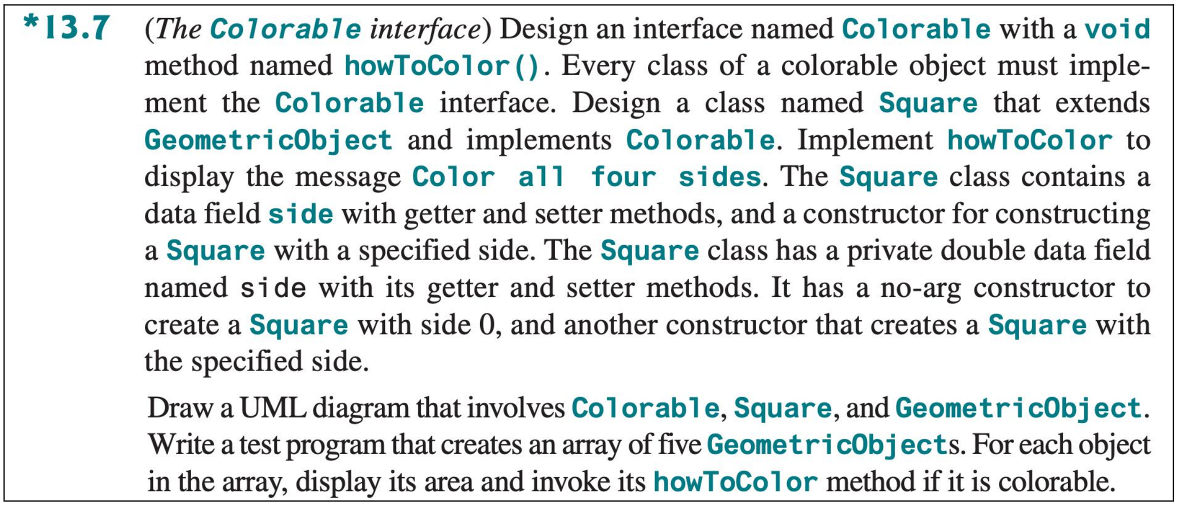


```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java ...
Enter the properties of a triangle:
Enter side1 length: 3
Enter side2 length: 4
Enter side3 length: 5
Enter its color: red
Enter whether the triangle is filled: false
The triangle's area is 6.0, its perimeter is 12.0, created on Sun May 28 09:55:57 CST 2023
color: red and filled: false

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: The isTriangle method is modified, but this method is used before the side variable is assigned, and it is judged that it cannot form a triangle.
Fix: Use the debugger to debug and check errors step by step, and locate the problem referenced before the assignment and modify it.

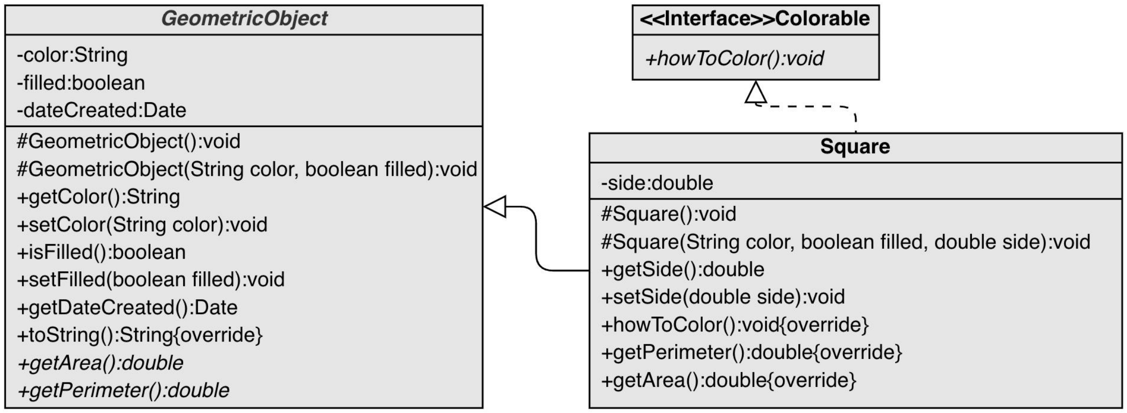
Problem 7. (13.7)



***13.7** (*The* `Colorable` *interface*) Design an interface named `Colorable` with a `void` method named `howToColor()`. Every class of a colorable object must implement the `Colorable` interface. Design a class named `Square` that extends `GeometricObject` and implements `Colorable`. Implement `howToColor` to display the message `Color all four sides`. The `Square` class contains a data field `side` with getter and setter methods, and a constructor for constructing a `Square` with a specified side. The `Square` class has a private double data field named `side` with its getter and setter methods. It has a no-arg constructor to create a `Square` with side 0, and another constructor that creates a `Square` with the specified side.

Draw a UML diagram that involves `Colorable`, `Square`, and `GeometricObject`. Write a test program that creates an array of five `GeometricObject`s. For each object in the array, display its area and invoke its `howToColor` method if it is colorable.

**** Source Code / Solution :***


GeometricObject
-color:String
-filled:boolean
-dateCreated:Date
#GeometricObject():void
#GeometricObject(String color, boolean filled):void
+getColor():String
+setColor(String color):void
+isFilled():boolean
+setFilled(boolean filled):void
+getDateCreated():Date
+toString():String{override}
+getArea():double
+getPerimeter():double
<<Interface>>Colorable
+howToColor():void
Square
-side:double
#Square():void
#Square(String color, boolean filled, double side):void
+getSide():double
+setSide(double side):void
+howToColor():void{override}
+getPerimeter():double{override}
+getArea():double{override}

```
public interface Colorable {
  void howToColor();
}
```

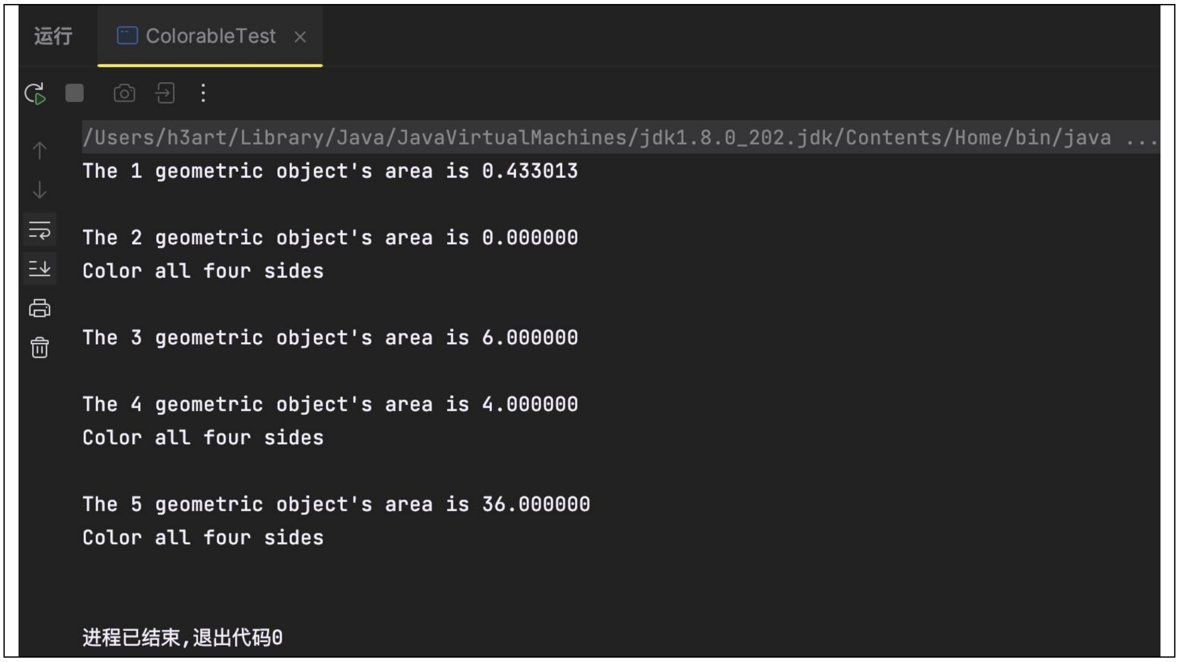
```
public abstract class GeometricObject {
  private String color = "white";
  private boolean filled;
  private java.util.Date dateCreated;

  /**
   * Construct a default geometric object
   */
  protected GeometricObject() {
     dateCreated = new java.util.Date();
  }

  /**
   * Construct a geometric object with color and filled value
   */
  protected GeometricObject(String color, boolean filled) {
     dateCreated = new java.util.Date();
     this.color = color;
     this.filled = filled;
  }

  /**
   * Return color
   */
  public String getColor() {
     return color;
  }



  /**
   * Set a new color
   */
  public void setColor(String color) {
     this.color = color;
  }



  /**
   * Return filled. Since filled is boolean,
   * 29 * the getter method is named isFilled
   */
  public boolean isFilled() {
     return filled;
  }

  /**
   * Set a new filled
   */
  public void setFilled(boolean filled) {
     this.filled = filled;
  }

  /**
   * Get dateCreated
   */
  public java.util.Date getDateCreated() {
     return dateCreated;

  }

  @Override
  public String toString() {
     return "created on " + dateCreated + "\ncolor: " + color +
             " and filled: " + filled;
  }
```

```java
  /**
   * Abstract method getArea
   */
  public abstract double getArea();

  /**
   * Abstract method getPerimeter
   */
  public abstract double getPerimeter();
}
```

```java
public class Square extends GeometricObject implements Colorable {
  private double side;

  protected Square() {
      super();
      this.side = 0;
  }

  protected Square(String color, boolean filled, double side) {
      super(color, filled);
      this.side = side;
  }

  public double getSide() {
      return side;
  }

  public void setSide(double side) {
      this.side = side;
  }
  @Override
  public void howToColor() {
      System.out.println("Color all four sides");
  }

  @Override
  public double getPerimeter() {
      return 4 * side;
  }

  @Override
  public double getArea() {
    return side * side;
  }
}
```

```
public class ColorableTest {
  public static void main(String[] args) {
    GeometricObject[] geometricObjectsArray = new GeometricObject[5];

    geometricObjectsArray[0] = new Triangle();
    geometricObjectsArray[1] = new Square();
    geometricObjectsArray[2] = new Triangle("blue", true, 3, 4, 5);
    geometricObjectsArray[3] = new Square("red", false, 2);
    geometricObjectsArray[4] = new Square("green", false, 6);

    for (int i = 0; i < 5; i++) {
      System.out.printf("The %d geometric object's area is %f\n", i + 1,
geometricObjectsArray[i].getArea());
      if (geometricObjectsArray[i] instanceof Colorable) {
        ((Colorable) geometricObjectsArray[i]).howToColor();
      }
      System.out.println();
    }
  }
}
```

**** Output:***

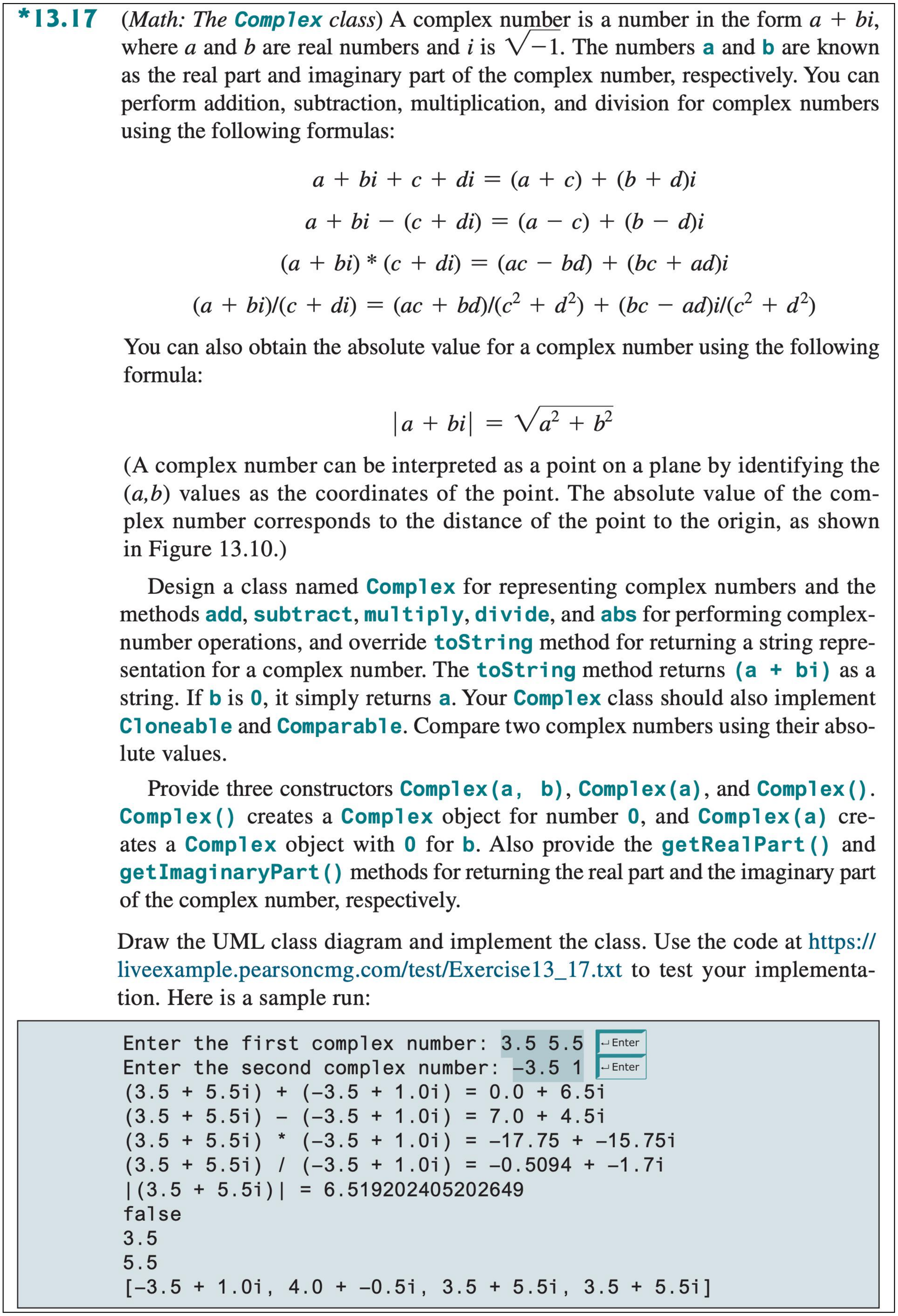


```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java ...
The 1 geometric object's area is 0.433013

The 2 geometric object's area is 0.000000
Color all four sides

The 3 geometric object's area is 6.000000

The 4 geometric object's area is 4.000000
Color all four sides

The 5 geometric object's area is 36.000000
Color all four sides


进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: Before using mandatory type conversion (awakening the polymorphism of the subclass to the parent class), the method of the subclass is called on the instance of the declared parent class object, and the compilation fails.
Fix: When using the method of the corresponding subclass on the basis of the parent class instance, it needs to be enclosed in parentheses, and cast to it within the parentheses to awaken its polymorphism.

<u>Problem 8. (13.17)(Optional)</u>



***13.17** (*Math: The* **`Complex`** *class*) A complex number is a number in the form $a + bi$, where a and b are real numbers and i is $\sqrt{-1}$. The numbers **`a`** and **`b`** are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formulas:

$$a + bi + c + di = (a + c) + (b + d)i$$

$$a + bi - (c + di) = (a - c) + (b - d)i$$

$$(a + bi) * (c + di) = (ac - bd) + (bc + ad)i$$

$$(a + bi)/(c + di) = (ac + bd)/(c^2 + d^2) + (bc - ad)i/(c^2 + d^2)$$

You can also obtain the absolute value for a complex number using the following formula:

$$|a + bi| = \sqrt{a^2 + b^2}$$

(A complex number can be interpreted as a point on a plane by identifying the (a,b) values as the coordinates of the point. The absolute value of the complex number corresponds to the distance of the point to the origin, as shown in Figure 13.10.)

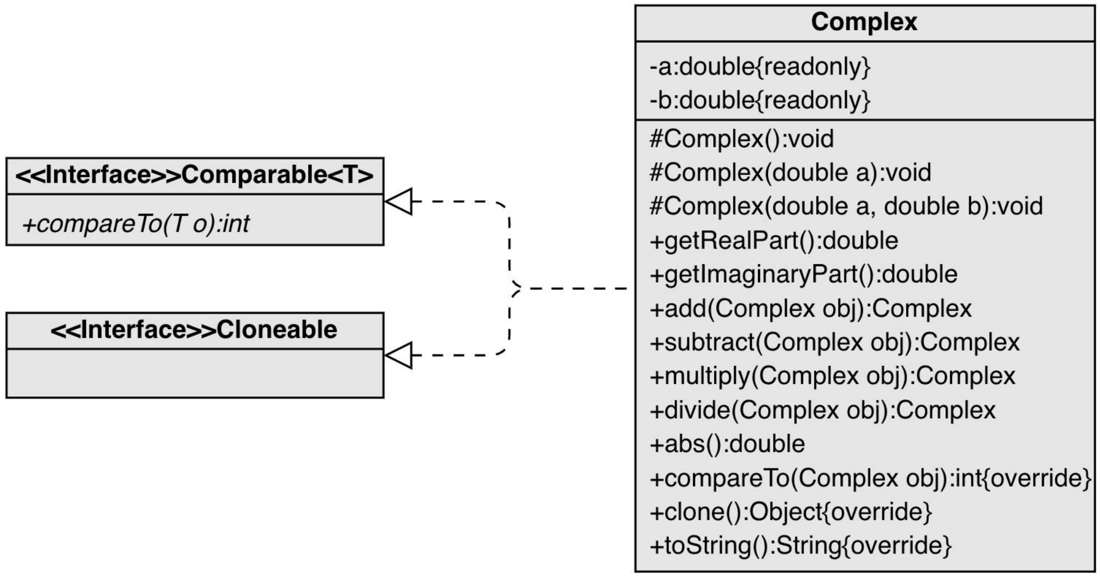
Design a class named **`Complex`** for representing complex numbers and the methods **`add`**, **`subtract`**, **`multiply`**, **`divide`**, and **`abs`** for performing complex-number operations, and override **`toString`** method for returning a string representation for a complex number. The **`toString`** method returns **`(a + bi)`** as a string. If **`b`** is **`0`**, it simply returns **`a`**. Your **`Complex`** class should also implement **`Cloneable`** and **`Comparable`**. Compare two complex numbers using their absolute values.

Provide three constructors **`Complex(a, b)`**, **`Complex(a)`**, and **`Complex()`**. **`Complex()`** creates a **`Complex`** object for number **`0`**, and **`Complex(a)`** creates a **`Complex`** object with **`0`** for **`b`**. Also provide the **`getRealPart()`** and **`getImaginaryPart()`** methods for returning the real part and the imaginary part of the complex number, respectively.

Draw the UML class diagram and implement the class. Use the code at https://liveexample.pearsoncmg.com/test/Exercise13_17.txt to test your implementation. Here is a sample run:

```
Enter the first complex number: 3.5 5.5 ↵ Enter
Enter the second complex number: -3.5 1 ↵ Enter
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -15.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i
|(3.5 + 5.5i)| = 6.519202405202649
false
3.5
5.5
[-3.5 + 1.0i, 4.0 + -0.5i, 3.5 + 5.5i, 3.5 + 5.5i]
```

**** Source Code / Solution :***

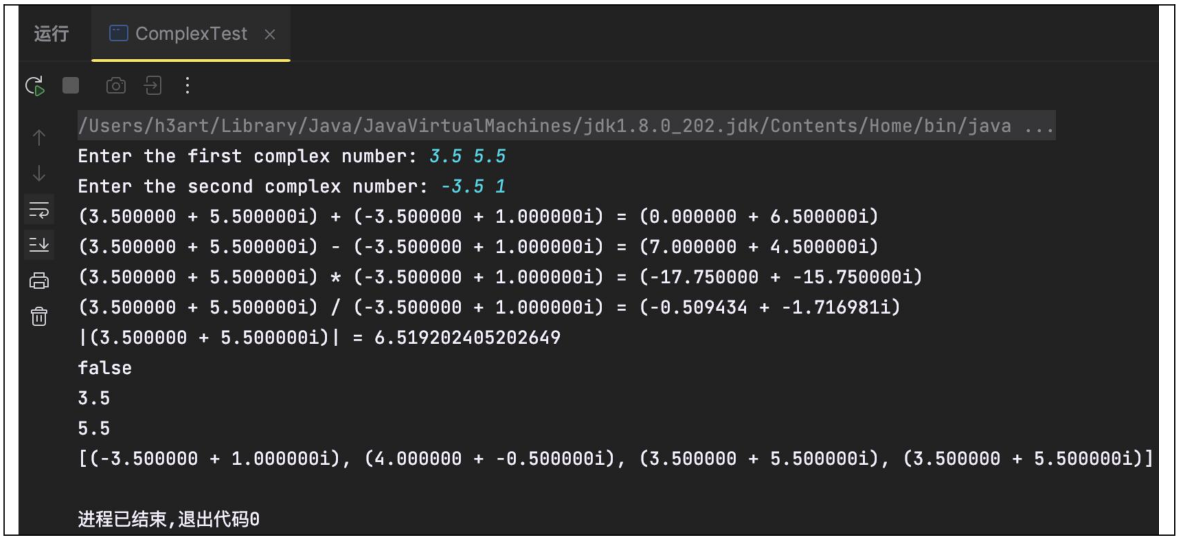

Complex
-a:double{readonly}
-b:double{readonly}
#Complex():void
#Complex(double a):void
#Complex(double a, double b):void
+getRealPart():double
+getImaginaryPart():double
+add(Complex obj):Complex
+subtract(Complex obj):Complex
+multiply(Complex obj):Complex
+divide(Complex obj):Complex
+abs():double
+compareTo(Complex obj):int{override}
+clone():Object{override}
+toString():String{override}
<<Interface>>Comparable<T>
+compareTo(T o):int
<<Interface>>Cloneable

```java
public class Complex implements Comparable<Complex>, Cloneable {
  private final double a;
  private final double b;

  protected Complex() {
    this.a = 0;
    this.b = 0;
  }

  protected Complex(double a) {
    this.a = a;
    this.b = 0;
  }

  protected Complex(double a, double b) {
    this.a = a;
    this.b = b;
  }

  public double getRealPart() {
    return a;
  }
```

```
    public double getImaginaryPart() {
        return b;
    }

    public Complex add(Complex obj) {
        return new Complex(a + obj.a, b + obj.b);
    }

    public Complex subtract(Complex obj) {
        return new Complex(a - obj.a, b - obj.b);
    }

    public Complex multiply(Complex obj) {
        return new Complex(a * obj.a - b * obj.b, b * obj.a + a * obj.b);
    }

    public Complex divide(Complex obj) {
        return new Complex(
                (a * obj.a + b * obj.b) / (obj.a * obj.a + obj.b * obj.b),
                (b * obj.a - a * obj.b) / (obj.a * obj.a + obj.b * obj.b)
        );
    }

    public double abs() {
        return Math.sqrt(a * a + b * b);
    }

    @Override
    public int compareTo(Complex obj) {
        return Double.compare(this.abs(), obj.abs());
    }

    @Override
    public Object clone() {
        try {
            return super.clone();
        } catch (CloneNotSupportedException e) {
            return null;
        }
    }

    @Override
    public String toString() {
        if (b != 0)
            return String.format("(%f + %fi)", a, b);
        else
            return String.valueOf(a);
    }
}
```

```
import java.util.Scanner;

public class ComplexTest {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the first complex number: ");
        double a = input.nextDouble();
        double b = input.nextDouble();
        Complex c1 = new Complex(a, b);

        System.out.print("Enter the second complex number: ");
        double c = input.nextDouble();
        double d = input.nextDouble();
        Complex c2 = new Complex(c, d);

        System.out.println(c1 + " + " + c2 + " = " + c1.add(c2));
        System.out.println(c1 + " - " + c2 + " = " + c1.subtract(c2));
        System.out.println(c1 + " * " + c2 + " = " + c1.multiply(c2));
        System.out.println(c1 + " / " + c2 + " = " + c1.divide(c2));
        System.out.println("|" + c1 + "| = " + c1.abs());

        Complex c3 = (Complex) c1.clone();
        System.out.println(c1 == c3);
        System.out.println(c3.getRealPart());
        System.out.println(c3.getImaginaryPart());

        Complex c4 = new Complex(4, -0.5);
        Complex[] list = {c1, c2, c3, c4};
        java.util.Arrays.sort(list);
        System.out.println(java.util.Arrays.toString(list));
    }
}
```

**** Output:***



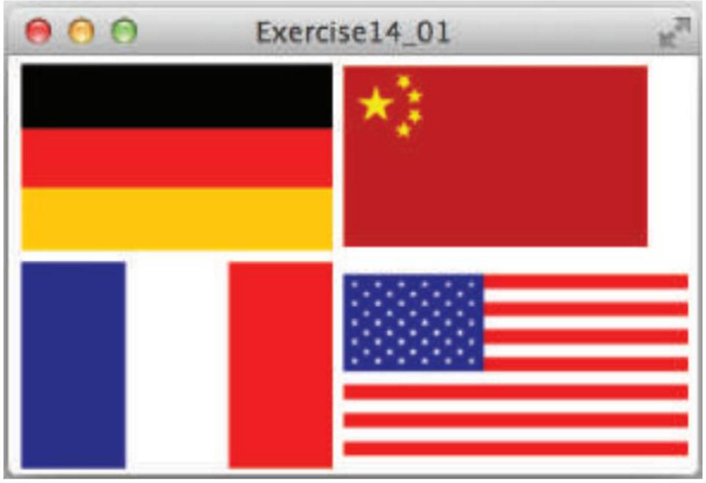
```
/Users/h3art/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java ...
Enter the first complex number: 3.5 5.5
Enter the second complex number: -3.5 1
(3.500000 + 5.500000i) + (-3.500000 + 1.000000i) = (0.000000 + 6.500000i)
(3.500000 + 5.500000i) - (-3.500000 + 1.000000i) = (7.000000 + 4.500000i)
(3.500000 + 5.500000i) * (-3.500000 + 1.000000i) = (-17.750000 + -15.750000i)
(3.500000 + 5.500000i) / (-3.500000 + 1.000000i) = (-0.509434 + -1.716981i)
|(3.500000 + 5.500000i)| = 6.519202405202649
false
3.5
5.5
[(-3.500000 + 1.000000i), (4.000000 + -0.500000i), (3.500000 + 5.500000i), (3.500000 + 5.500000i)]

进程已结束,退出代码0
```

**** Debugging/Testing:***

Bug1: According to the given test file, it does not handle the exception thrown by the clone method of the Complex class, so it is necessary to modify the implementation of the clone method in the Complex class.
Fix: When overriding the clone method, place CloneNotSupportedException in the overridden clone method and use a try-catch block for processing, and return a null value if an exception is triggered.

Problem 9. (14.1)

14.1 (*Display images*) Write a program that displays four images in a grid pane, as shown in Figure 14.43a.



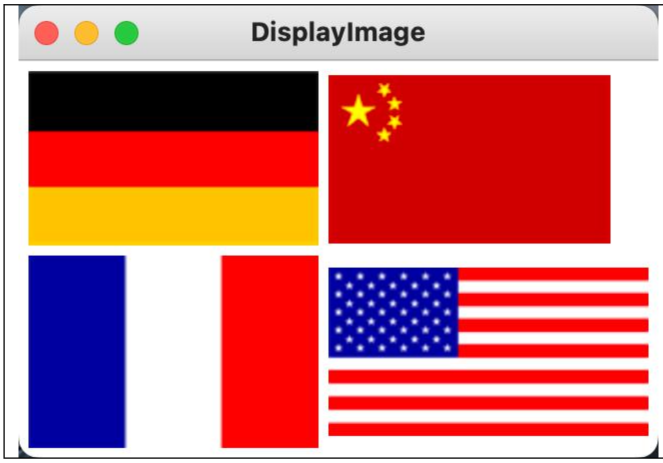
(a)

**** Source Code / Solution :***

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.GridPane;
import javafx.geometry.Insets;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;

public class DisplayImages extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
    // Create a pane to hold the image views
    GridPane totalPane = new GridPane();
    totalPane.setPadding(new Insets(5, 5, 5, 5));
    totalPane.setHgap(5);
    totalPane.setVgap(5);

    Image image1 = new Image("image/germany.gif");
    Image image2 = new Image("image/china.gif");
    Image image3 = new Image("image/fr.gif");
    Image image4 = new Image("image/us.gif");

    totalPane.add(new ImageView(image1), 0, 0);
```

```
        totalPane.add(new ImageView(image2), 1, 0);
        totalPane.add(new ImageView(image3), 0, 1);
        totalPane.add(new ImageView(image4), 1, 1);

        // Create a scene and place it in the stage
        Scene scene = new Scene(totalPane);
        primaryStage.setTitle("DisplayImage"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }
}
```

**** Output:***



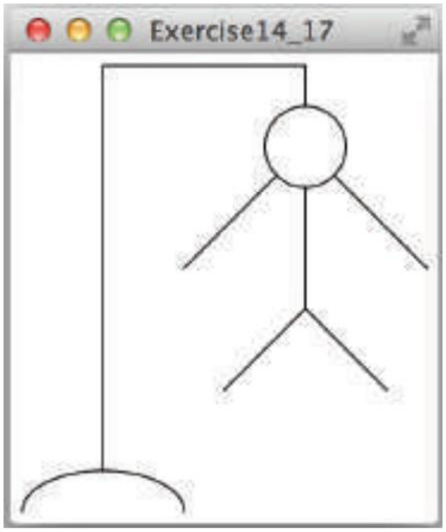
**** Debugging/Testing:***

Bug1: If a regular Pane is used as the display panel, the display of multiple images will overlap and cannot output the required effect.

Fix: Using GridPane as the display panel and adding images to different corners can display the required effect.

Problem 10. (14.17)

14.17 (*Game: hangman*) Write a program that displays a drawing for the popular hangman game, as shown in Figure 14.48a.

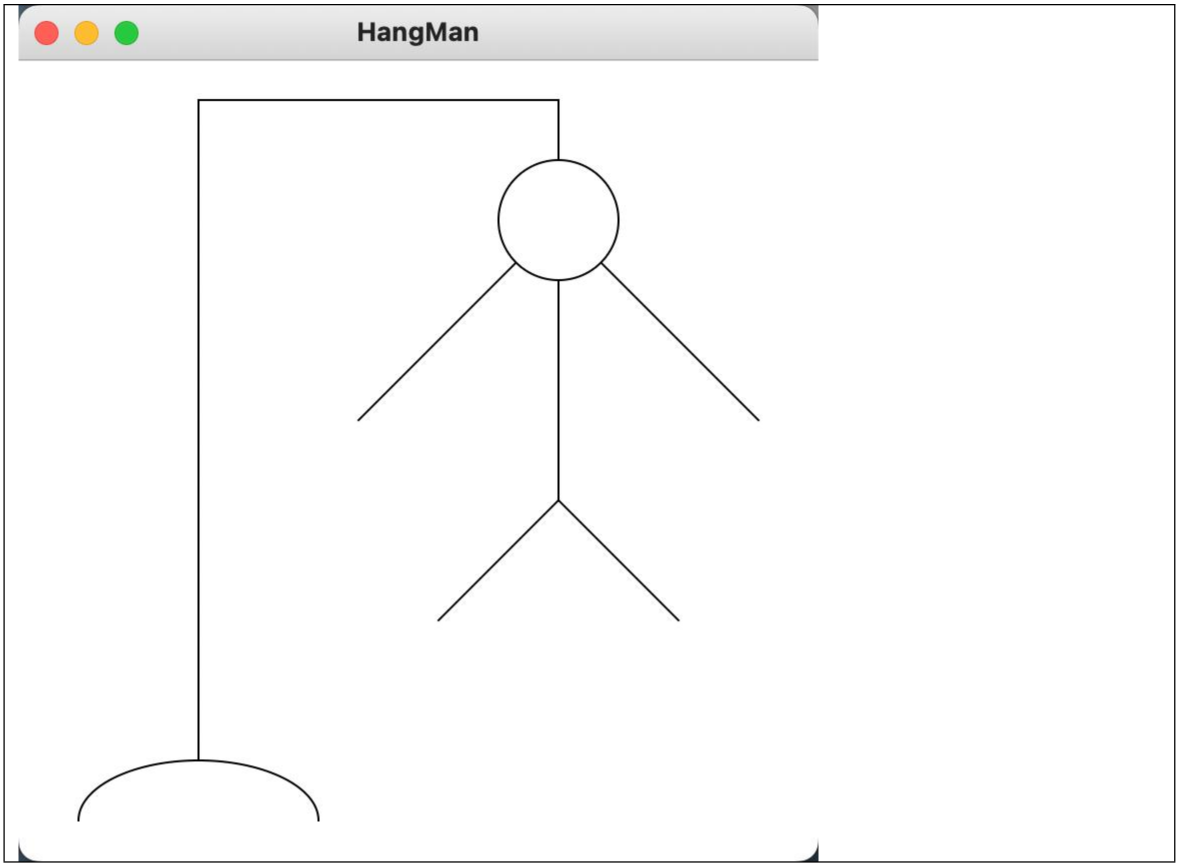


(a)

**** Source Code / Solution :***

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Arc;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.stage.Stage;

// 14.17
public class HangMan extends Application {
    @Override
    public void start(Stage primaryStage){
        Arc arc = new Arc(80, 380, 60, 30, 0, 180);
        arc.setFill(Color.WHITE);
        arc.setStroke(Color.BLACK);

        Line longVerticalLine = new Line(80, 350, 80, 20);
        Line longHorizontalLine = new Line(80, 20, 260, 20);
        Line shortVerticalLine = new Line(260, 20, 260, 50);

        Circle head = new Circle(260, 80, 30, Color.WHITE);
        head.setStroke(Color.BLACK);

        Line body = new Line(260, 50, 260, 220);

        Line rightHand = new Line(260, 80, 160, 180);
        Line leftHand = new Line(260, 80, 360, 180);
        Line rightLeg = new Line(260, 220, 200, 280);
        Line leftLeg = new Line(260, 220, 320, 280);

        Group allElement = new Group();
        allElement.getChildren().addAll(
                arc,
                longVerticalLine,
                longHorizontalLine,
                shortVerticalLine,
                body,
                rightHand,
                leftHand,
                rightLeg,
                leftLeg,
                head
        );

        // Create a scene and place it in the stage
        Scene scene = new Scene(new BorderPane(allElement), 400, 400);
        primaryStage.setTitle("HangMan"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }
}
```

**** Output:***

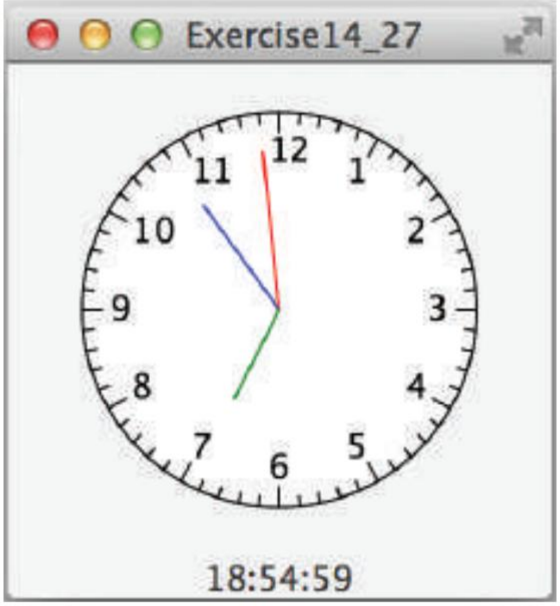


**** Debugging/Testing:***

Bug1: The position of the line drawn is not accurate.
Fix: Continuously compare and adjust, and finally present a pattern that is similar to the requirements.

Problem 11. (14.27)(Optional)

***14.27** (*Draw a detailed clock*) Modify the `ClockPane` class in Section 14.12 to draw the clock with more details on the hours and minutes, as shown in Figure 14.52a.



(a)

**** Source Code / Solution :***

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.scene.text.Text;
import javafx.stage.Stage;

import java.util.Calendar;
import java.util.GregorianCalendar;

public class DetailedClock extends Application {
  @Override // Override the start method in the Application class
  public void start(Stage primaryStage) {
    // Create a clock
    ClockPane clock = new ClockPane();
    // Create a label
```

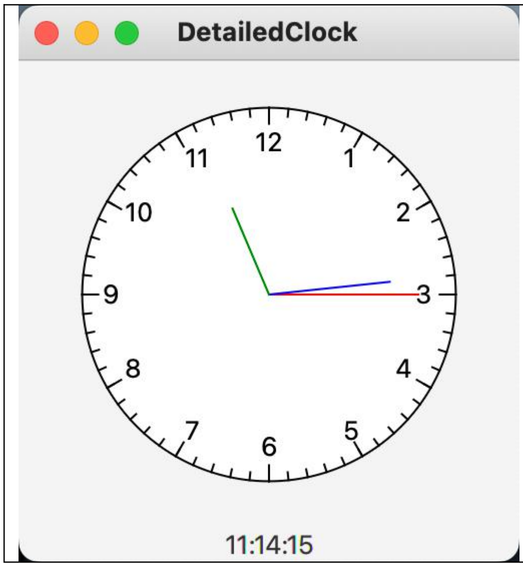
```
        String timeString = String.format("%02d:%02d:%02d"
                , clock.getHour()
                , clock.getMinute()
                , clock.getSecond());
        Label lblCurrentTime = new Label(timeString);

        // Place clock in center of border pane
        BorderPane pane = new BorderPane();
        pane.setCenter(clock);

        // Place label in bottom of border pane, and set alignment
        pane.setBottom(lblCurrentTime);
        BorderPane.setAlignment(lblCurrentTime, Pos.TOP_CENTER);

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane, 250, 250);
        primaryStage.setTitle("DetailedClock"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }

}


class ClockPane extends Pane {
    private int hour;
    private int minute;
    private int second;

    /**
     * Construct a default clock with the current time
     */
    public ClockPane() {
        setCurrentTime();
    }

    /**
     * Construct a clock with specified hour, minute, and second
     */
```

```java
    public ClockPane(int hour, int minute, int second) {
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    /**
     * Return hour
     */
    public int getHour() {
        return hour;
    }

    /**
     * Set a new hour
     */
    public void setHour(int hour) {
        this.hour = hour;
        paintClock();
    }

    /**
     * Return minute
     */
    public int getMinute() {
        return minute;
    }

    /**
     * Set a new minute
     */
    public void setMinute(int minute) {
        this.minute = minute;
        paintClock();
    }

    /**
     * Return second
     */
    public int getSecond() {
        return second;
    }

    /**
     * Set a new second
     */
    public void setSecond(int second) {
        this.second = second;
        paintClock();
    }

    /* Set the current time for the clock */
    public void setCurrentTime() {
        // Construct a calendar for the current date and time
        Calendar calendar = new GregorianCalendar();
        // Set current hour, minute and second
        this.hour = calendar.get(Calendar.HOUR_OF_DAY);
        this.minute = calendar.get(Calendar.MINUTE);
        this.second = calendar.get(Calendar.SECOND);
        paintClock(); // Repaint the clock
    }

    /**
     * Paint the total clock
     */
    private void paintClock() {
        // Initialize clock parameters
        double clockRadius = Math.min(getWidth(), getHeight()) * 0.8 * 0.5;
        double centerX = getWidth() / 2;
        double centerY = getHeight() / 2;
        getChildren().clear();

        // Draw circle
        Circle circle = new Circle(centerX, centerY, clockRadius);
        circle.setFill(Color.WHITE);
        circle.setStroke(Color.BLACK);
        getChildren().add(circle);
    // Draw hour label
    paintLabel(centerX, centerY, clockRadius);

    // Draw long calibration
    paintLongCalibration(centerX, centerY, clockRadius);

    // Draw short calibration
    paintShortCalibration(centerX, centerY, clockRadius);

    // Draw clock hand
    paintClockHand(centerX, centerY, clockRadius);
}

/**
 * Paint the short calibration of clock
 */
private void paintShortCalibration(double centerX, double centerY, double clockRadius) {
    double startRadius = clockRadius - 4;
    for (int i = 0; i < 60; i++) {
        double startX = centerX + startRadius *
                Math.sin(i * (2 * Math.PI / 60));
        double startY = centerY + startRadius *
                Math.cos(i * (2 * Math.PI / 60));
        double endX = centerX + clockRadius *
                Math.sin(i * (2 * Math.PI / 60));
        double endY = centerY + clockRadius *
                Math.cos(i * (2 * Math.PI / 60));
        Line calibration = new Line(startX, startY, endX, endY);
        getChildren().add(calibration);
    }
}

/**
 * Paint the long calibration of clock
 */
private void paintLongCalibration(double centerX, double centerY, double clockRadius) {
    double startRadius = clockRadius - 8;
    for (int i = 0; i < 12; i++) {
        double startX = centerX + startRadius *
```

```
                    Math.sin(i * (2 * Math.PI / 12));
            double startY = centerY + startRadius *
                    Math.cos(i * (2 * Math.PI / 12));
            double endX = centerX + clockRadius *
                    Math.sin(i * (2 * Math.PI / 12));
            double endY = centerY + clockRadius *
                    Math.cos(i * (2 * Math.PI / 12));
            Line calibration = new Line(startX, startY, endX, endY);
            getChildren().add(calibration);
        }
    }

    /**
     * Paint the clock hand
     */
    private void paintClockHand(double centerX, double centerY, double clockRadius) {
        // Draw second hand
        double sLength = clockRadius * 0.8;
        double secondX = centerX + sLength *
                Math.sin(second * (2 * Math.PI / 60));
        double secondY = centerY - sLength *
                Math.cos(second * (2 * Math.PI / 60));
        Line sLine = new Line(centerX, centerY, secondX, secondY);
        sLine.setStroke(Color.RED);

        // Draw minute hand
        double mLength = clockRadius * 0.65;
        double minuteX = centerX + mLength *
                Math.sin(minute * (2 * Math.PI / 60));
        double minuteY = centerY - mLength *
                Math.cos(minute * (2 * Math.PI / 60));
        Line mLine = new Line(centerX, centerY, minuteX, minuteY);
        mLine.setStroke(Color.BLUE);

        // Draw hour hand
        double hLength = clockRadius * 0.5;
        double hourX = centerX + hLength *
                Math.sin((hour % 12 + minute / 60.0) * (2 * Math.PI / 12));
        double hourY = centerY - hLength *
```

```
                Math.cos((hour % 12 + minute / 60.0) * (2 * Math.PI / 12));
        Line hLine = new Line(centerX, centerY, hourX, hourY);
        hLine.setStroke(Color.GREEN);

        getChildren().addAll(sLine, mLine, hLine);
    }

    /**
     * Paint hour label
     */
    private void paintLabel(double centerX, double centerY, double clockRadius) {
        Text t1 = new Text(centerX + (clockRadius - 20) / 2,
                centerY - (clockRadius - 20) * Math.sqrt(3) / 2, "1");
        Text t2 = new Text(centerX + (clockRadius - 20) * Math.sqrt(3) / 2,
                centerY - (clockRadius - 20) / 2, "2");
        Text t3 = new Text(centerX + clockRadius - 20, centerY + 4, "3");

        Text t4 = new Text(centerX + (clockRadius - 20) * Math.sqrt(3) / 2,
                centerY + (clockRadius - 10) / 2, "4");
        Text t5 = new Text(centerX + (clockRadius - 20) / 2,
                centerY + (clockRadius - 10) * Math.sqrt(3) / 2, "5");
        Text t6 = new Text(centerX - 4, centerY + clockRadius - 13, "6");

        Text t7 = new Text(centerX - (clockRadius - 10) / 2,
                centerY + (clockRadius - 10) * Math.sqrt(3) / 2, "7");
        Text t8 = new Text(centerX - (clockRadius - 10) * Math.sqrt(3) / 2,
                centerY + (clockRadius - 10) / 2, "8");
        Text t9 = new Text(centerX - clockRadius + 10, centerY + 4, "9");

        Text t10 = new Text(centerX - (clockRadius - 10) * Math.sqrt(3) / 2,
                centerY - (clockRadius - 20) / 2, "10");
        Text t11 = new Text(centerX - (clockRadius - 10) / 2,
                centerY - (clockRadius - 20) * Math.sqrt(3) / 2, "11");
        Text t12 = new Text(centerX - 7, centerY - clockRadius + 22, "12");

        getChildren().addAll(t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12);
    }

    @Override
    public void setWidth(double width) {
        super.setWidth(width);
        paintClock();
    }

    @Override
    public void setHeight(double height) {
        super.setHeight(height);
        paintClock();
    }
}
```

**** Output:***



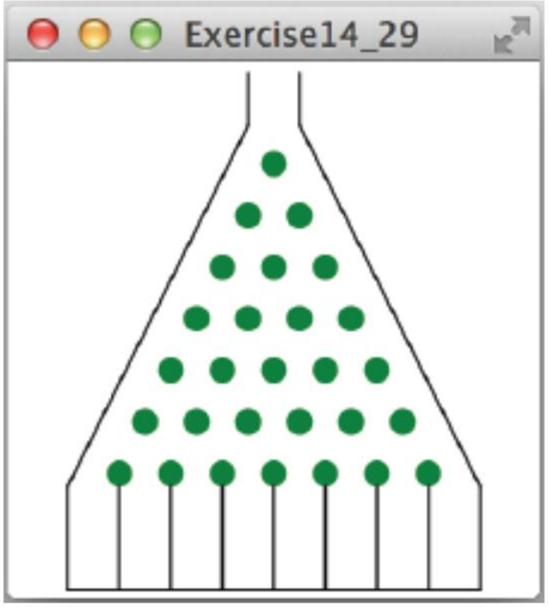
11:14:15

**** Debugging/Testing:***

```
Bug1: I wanted to use a loop to directly draw the time
characters on the dial, but in the end, all of them were
misplaced.
Fix: In the end, by separately drawing and adjusting the
position of each scale mark, a good effect was achieved.
```

Problem 12. (14.29)(Optional)

****14.29** (*Game: bean machine*) Write a program that displays a bean machine introduced in Programming Exercise 7.37, as shown in Figure 14.52c.

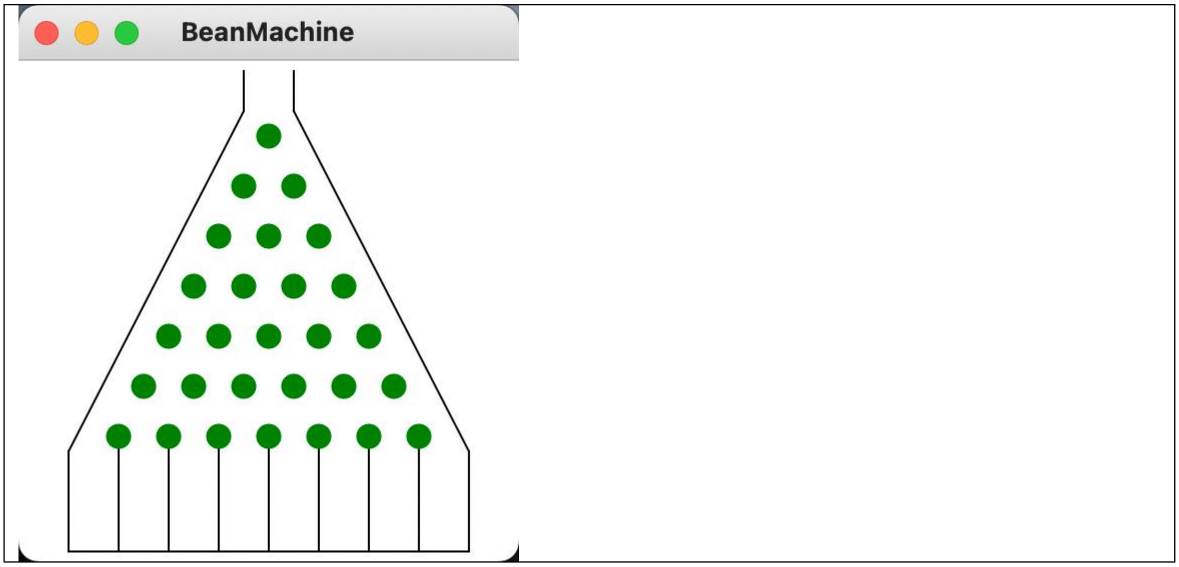


(c)

**** Source Code / Solution :***

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.stage.Stage;

public class BeanMachine extends Application {
    @Override
    public void start(Stage primaryStage) {
        BorderPane pane = new BorderPane();
        BeanPane beanMachine = new BeanPane();

        pane.setCenter(beanMachine);

        // Create a scene and place it in the stage
        Scene scene = new Scene(pane, 250, 250);
        primaryStage.setTitle("BeanMachine"); // Set the stage title
        primaryStage.setScene(scene); // Place the scene in the stage
        primaryStage.show(); // Display the stage
    }
}

class BeanPane extends Pane {
    BeanPane() {
        paint();
    }

    public void paint() {
        double topCenterPointX = getWidth() / 2;
        double topCenterPointY = 0;
        getChildren().clear();

        paintFrame(topCenterPointX, topCenterPointY);

        paintCircles(topCenterPointX, topCenterPointY);
    }

    private void paintCircles(double topCenterPointX, double topCenterPointY) {
        double circleGap = getWidth() / 20;
        double circleRadius = circleGap / 2;
        double startX = topCenterPointX;
        double startY = topCenterPointY + circleGap + getWidth() / 10;

        for (int i = 0; i < 7; i++) {
            for (int j = 0; j <= i; j++) {
                Circle circle = new Circle(startX + (2 * circleGap) * j, startY, circleRadius);
                circle.setFill(Color.GREEN);
                getChildren().add(circle);
            }
            startX -= circleGap;
            startY += 2 * circleGap;
        }
    }

    private void paintFrame(double topCenterPointX, double topCenterPointY) {
        Line leftEntrance = new Line(
                topCenterPointX - getWidth() / 20,
                topCenterPointY + 5,
                topCenterPointX - getWidth() / 20,
                topCenterPointY + getWidth() / 10
        );
        Line rightEntrance = new Line(
                topCenterPointX + getWidth() / 20,
                topCenterPointY + 5,
                topCenterPointX + getWidth() / 20,
                topCenterPointY + getWidth() / 10
        );
        Line leftHypotenuse = new Line(
                topCenterPointX - getWidth() / 20,
                topCenterPointY + getWidth() / 10,
                topCenterPointX - 0.4 * getWidth(),
                topCenterPointY + 0.78 * getWidth()

        );
        Line rightHypotenuse = new Line(
                topCenterPointX + getWidth() / 20,
                topCenterPointY + getWidth() / 10,
                topCenterPointX + 0.4 * getWidth(),
                topCenterPointY + 0.78 * getWidth()
        );
        Line leftBottomLine = new Line(
                topCenterPointX - 0.4 * getWidth(),
                topCenterPointY + 0.78 * getWidth(),
                topCenterPointX - 0.4 * getWidth(),
                topCenterPointY + 0.98 * getWidth()
        );
        Line rightBottomLine = new Line(
                topCenterPointX + 0.4 * getWidth(),
                topCenterPointY + 0.78 * getWidth(),
                topCenterPointX + 0.4 * getWidth(),
                topCenterPointY + 0.98 * getWidth()
        );
        Line bottomLine = new Line(
                topCenterPointX - 0.4 * getWidth(),
                topCenterPointY + 0.98 * getWidth(),
                topCenterPointX + 0.4 * getWidth(),
                topCenterPointY + 0.98 * getWidth()
        );
        double startX = topCenterPointX - 0.3 * getWidth();
        for (int i = 0; i < 7; i++) {
            Line verticalSepLine = new Line(
                    startX,
                    topCenterPointY + 0.76 * getWidth(),
                    startX,
                    topCenterPointY + 0.98 * getWidth()
            );
            startX += 0.1 * getWidth();
            getChildren().add(verticalSepLine);
        }
        getChildren().addAll(
                leftEntrance,
                rightEntrance,
                leftHypotenuse,
                rightHypotenuse,
                leftBottomLine,
                rightBottomLine,
                bottomLine
        );
    }

    @Override
    public void setWidth(double width) {
        super.setWidth(width);
        paint();
    }

    @Override
    public void setHeight(double height) {
        super.setHeight(height);
        paint();
    }
}
```

**** Output:***



**** Debugging/Testing:***

Bug1: Fixed line coordinates were designed, but the internal green sphere was dynamically drawn, ultimately resulting in incorrect shape of the entire pattern if the window size was adjusted.
Fix: Set the coordinates of the line to dynamically adjust.