```java
1  package Ch2Task.E2_6;
2
3  /**
4   * Created by Sheldon on 10/23/2016.
5   * E2.6
6   * Replacing string object elements
7   */
8
9  public class HollePrinter {
10     public static void main(String[] args) {
11         // Construct string object
12         String sentence = "Hello, World!";
13
14         // 1st. replace string element e to $,  "H$llo, World!"
15         // 2nd. replace string element o to e,  "H$lle, Werld!"
16         // 3rd. replace string element $ to o,  "Holle, Werld!"
17         String sentenceReplace = sentence.replace("e", "$").replace("o", "e").replace("$", "o");
18
19         // Display the result
20         System.out.printf("String %25.15s%nString changed %17.15s%nExpected %23.15s%n",
21             ": " + sentence, ": " + sentenceReplace, ": " + "Holle, Werld!");
22     }
23  }
```
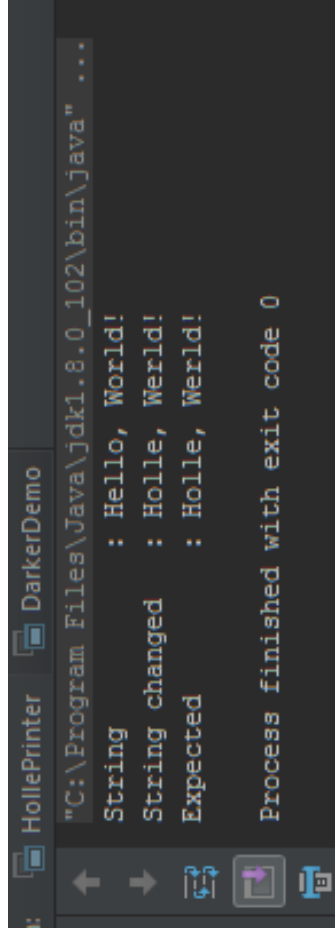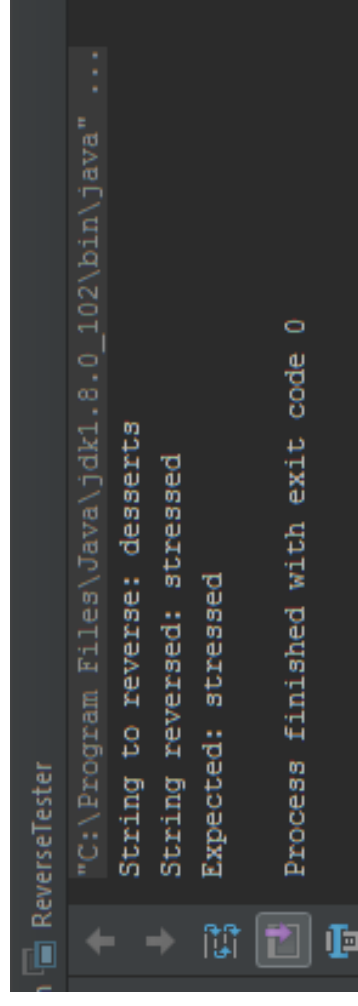
```
HollePrinter    DarkerDemo

"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...

String          : Hello, World!
String changed  : Holle, Werld!
Expected        : Holle, Werld!

Process finished with exit code 0
```

```java
1  package Ch2Task.E2_7;
2
3  /**
4   * Created by Sheldon on 10/23/2016.
5   * E2.7
6   * Reversing string object
7   */
8  public class ReverseTester {
9      public static void main(String[] args) {
10         // Construct string object
11         StringBuilder stringToReverse = new StringBuilder("desserts");
12         System.out.println("String to reverse: " + stringToReverse);
13
14         // Call reverse() method to reverse the string object
15         String stringReversed = stringToReverse.reverse().toString();
16
17         // Display result
18         System.out.println("String reversed: " + stringReversed);
19         System.out.println("Expected: stressed");
20     }
21 }
22
```

ReverseTester

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
String to reverse: desserts
String reversed: stressed
Expected: stressed

Process finished with exit code 0
```

```java
1  package Ch2Task.E2_8;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.8
9   * Compare two or more colors are brighter or darker
10  */
11
12  public class BrighterDemo {
13      public static void main(String[] args) {
14          // Construct first color object
15          Color theFirstColor = new Color(50, 100, 150);
16
17          // Construct first frame
18          JFrame firstFrame = new JFrame();
19          firstFrame.setSize(250, 250);
20          firstFrame.setTitle("First Color");
21          firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22          firstFrame.setVisible(true);
23
24          // Apply first color to first frame's background color
25          firstFrame.getContentPane().setBackground(theFirstColor);
26
27          // Construct second color object
28          Color theSecondColor = theFirstColor.brighter();
29
30          // Construct second frame
31          JFrame secondFrame = new JFrame();
```
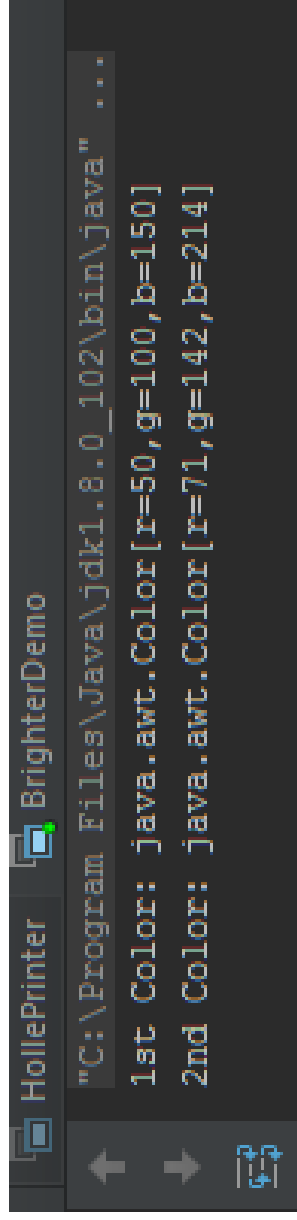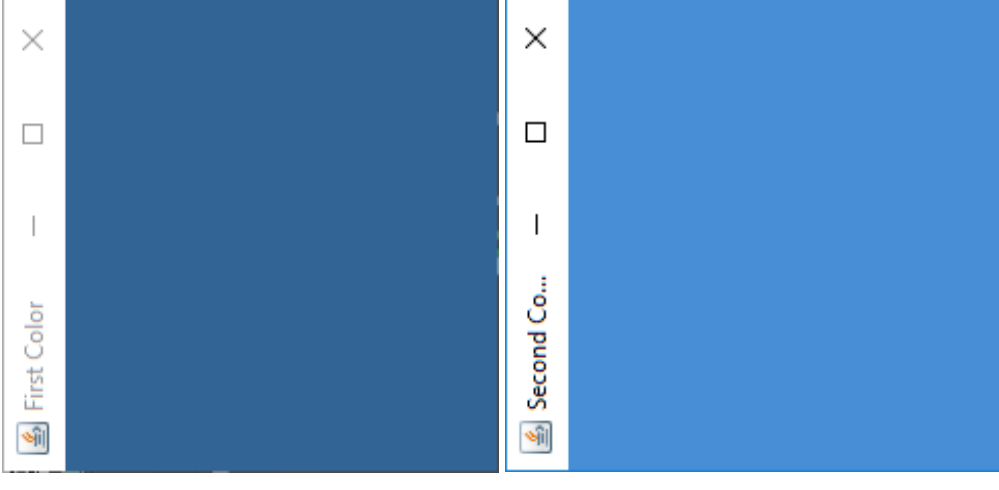
```
32    secondFrame.setSize(250, 250);
33    secondFrame.setTitle("Second Color");
34    secondFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35    secondFrame.setVisible(true);
36
37    // Apply second color to first frame's background color
38    secondFrame.getContentPane().setBackground(theSecondColor);
39
40    // Display the comparison between colors
41    System.out.println("1st Color: " + theFirstColor);
42    System.out.println("2nd Color: " + theSecondColor);
43    }
44  }
```



```
HollePrinter    BrighterDemo

"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...

1st Color: java.awt.Color[r=50,g=100,b=150]
2nd Color: java.awt.Color[r=71,g=142,b=214]
```

```java
1  package Ch2Task.P2_1;
2
3  import java.awt.Rectangle;
4  /**
5   * Created by Sheldon on 10/24/2016.
6   * P2.1
7   */
8  public class FourRectanglePrinter {
9      public static void main(String[] args) {
10         Rectangle box = new Rectangle(5, 10, 10, 20);
11         System.out.println(box);
12         box.translate(10, 0);
13         System.out.println(box);
14         box.translate(0, 20);
15         System.out.println(box);
16     }
17 }
18
```

Intersected Rectangles — □ ×

Window Snip

Intersection

```java
1  package Ch2Task.P2_3;
2
3  import javax.swing.JComponent;
4  import javax.swing.JFrame;
5  import java.awt.*;
6
7  /**
8   * Created by Sheldon on 10/27/2016.
9   * P2.3
10  * Decide wether intersection is occured or not
11  * Fill the intersection area if occured
12  */
13
14 public class IntersectionRectangle extends JComponent {
15     public static void main(String[] args) {
16         // Construct frame object
17         JFrame frame = new JFrame();
18         frame.setSize(600, 600);
19         frame.setTitle("Intersected Rectangles");
20         frame.getContentPane().setBackground(Color.WHITE);
21         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22         frame.setVisible(true);
23
24         // Construct rectangles
25         IntersectionRectangle intersectCmp = new IntersectionRectangle();
26
27         // Add the components to frame
28         frame.add(intersectCmp);
29     }
30     public void paintComponent(Graphics g) {
31         Graphics2D painter = (Graphics2D) g;    // Construct painter object
```
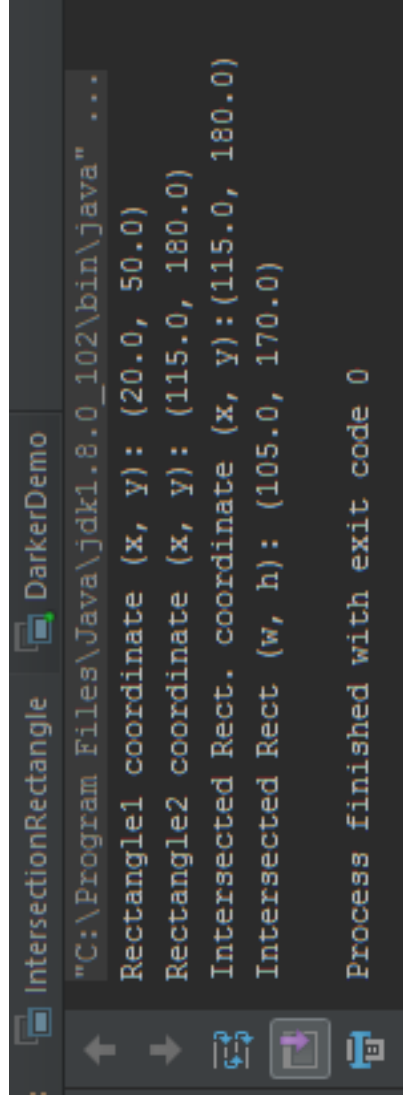
```
32
33    painter.setColor(Color.BLACK);
34    Rectangle r1 = new Rectangle(20, 50, 200, 300); // Construct first rectangle
35    painter.draw(r1);
36
37    painter.setColor(Color.BLUE);
38    Rectangle r2 = new Rectangle(115, 180, 300, 200); // Construct second rectangle
39    //   Rectangle r2 = new Rectangle(220, 350, 300, 200);
40    //   Rectangle r2 = new Rectangle(110, 350, 300, 200);
41    //   Rectangle r2 = new Rectangle(220, 175, 300, 200);
42    painter.draw(r2);
43
44    painter.setColor(Color.RED.darker());
45    Rectangle r3 = r1.intersection(r2); // Construct intersection rectangle
46    painter.fill(r3);
47    float xPoint = (float) r3.getX();
48    float yPoint = (float) r3.getY();
49
50    System.out.println("Rectangle1 coordinate (x, y): (" + r1.getX() + ", " + r1.getY() + ")");
51    System.out.println("Rectangle2 coordinate (x, y): (" + r2.getX() + ", " + r2.getY() + ")");
52
53    // if width and height is 0, no intersection happened
54    if (r3.getWidth() == 0 && r3.getHeight() == 0) {
55        System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
56        System.out.println("No intersection");
57    }
58    // if width is measured but height didn't, no intersection happened
59    else if (r3.getWidth() > 0 && r3.getHeight() == 0) {
60        System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
61        System.out.println("No intersection");
62    }
```

```java
63      // if height is measured but width didn't, no intersection happened
64      else if (r3.getWidth() == 0 && r3.getHeight() > 0) {
65          System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
66          System.out.println("No intersection");
67      }
68      // if width and height are measured, intersection happened
69      else if (r3.getWidth() > 0 && r3.getHeight() > 0) {
70          System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
71          System.out.println("Intersected Rect (w, h): (" + r3.getWidth() + ", " + r3.getHeight() + ")");
72          painter.setColor(Color.LIGHT_GRAY);
73          painter.drawString("Intersection", (xPoint + 10), (yPoint + 50));
74      }
75  }
76  }
77
78
```

IntersectionRectangle   DarkerDemo

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Rectangle1 coordinate (x, y): (20.0, 50.0)
Rectangle2 coordinate (x, y): (115.0, 180.0)
Intersected Rect. coordinate (x, y):(115.0, 180.0)
Intersected Rect (w, h): (105.0, 170.0)

Process finished with exit code 0
```

```
1  package Ch2Task.P2_5;
2
3  import java.util.Random;
4
5  /**
6   * Created by Sheldon on 10/24/2016.
7   * P2.5
8   * Generate lottery numbers array
9   */
10 public class LotteryPrinter {
11     public static void main(String[] args) {
12         int max = 50;
13         int offset = 1;
14         int range = max - offset;
15         int numbersToGenerate = 6;
16
17         Random generator = new Random();
18         System.out.println("Lottery combination: ");
19         for (int i = 0; i < numbersToGenerate; i++) {
20             int inRangeGenerator = generator.nextInt(range) + offset;
21             System.out.print(inRangeGenerator);
22             if (i < (numbersToGenerate - 1)) {
23                 System.out.print(" ");
24             }
25             if (i == (numbersToGenerate - 1)) {
26                 System.out.println("\nPlay this combination-it'll make you rich!");
27             }
28         }
29     }
30 }
31
```

LotteryPrinter   DarkerDemo

"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Lottery combination:
3 9 31 47 21 17
Play this combination-it'll make you rich!

Process finished with exit code 0

Line to Point Distance | □   ✕

Distance:141.42

Distance::0

Distance:70.71

```java
1  package Ch2Task.P2_9;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.geom.Line2D;
6  import java.text.DecimalFormat;
7
8  /**
9   * Created by Sheldon on 10/27/2016.
10  * P2.9
11  * Measure and draw distance from line segment to specified point
12  */
13 public class LineToPointDistance extends JComponent {
14     public static void main(String[] args) {
15         JFrame myFrame = new JFrame();
16         myFrame.setSize(400, 400);
17         myFrame.setTitle("Line to Point Distance");
18         myFrame.getContentPane().setBackground(Color.WHITE);
19         myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20         myFrame.setVisible(true);
21
22         LineToPointDistance calculateDistance = new LineToPointDistance();
23         myFrame.add(calculateDistance);
24     }
25     public void paintComponent(Graphics g) {
26         Graphics2D painter = (Graphics2D) g;        // Construct painter object
27         Line2D.Double lineSegment = new Line2D.Double(100, 100, 200, 200);  //Construct line object
28         painter.setColor(Color.BLUE);
29         painter.draw(lineSegment);
30
31         int[] x = {100, 150, 250};  // Create x ordinates list
```
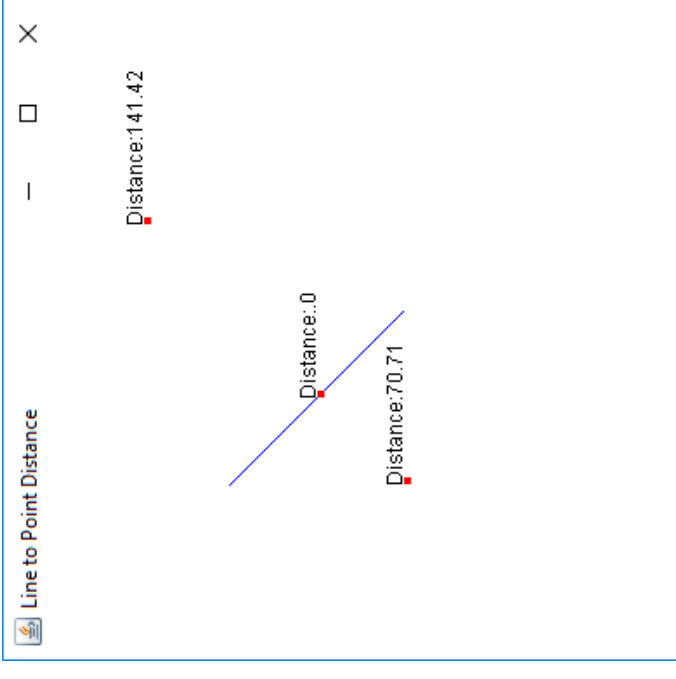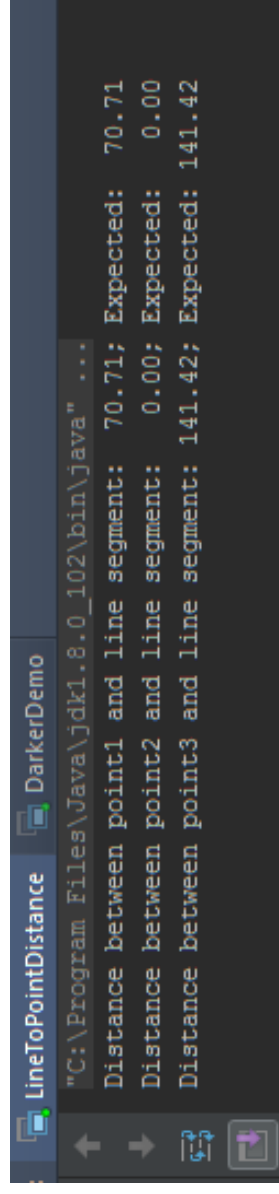
```java
32    int[] y = {200, 150, 50};       // Create y ordinates list
33    int dataLength = x.length;      // Hold data length
34    double[] toPointDistance = new double[dataLength];  // Create line-to-point-distance empty list
35    double[] expectation = {70.71, 0, 141.42};
36
37    for (int i = 0; i < dataLength; i++) {
38        // Fill oval object in coordinate (x[i], y[i]
39        painter.setColor(Color.RED);
40        painter.fillOval(x[i], y[i], 5, 5);
41
42        // Calculate distance from line segment to the point specified
43        toPointDistance[i] = lineSegment.ptSegDist(x[i], y[i]);
44        String pointDistanceFormatted = new DecimalFormat(".##").format(toPointDistance[i]);
45
46        // Draw string indicates the distance
47        painter.setColor(Color.BLACK);
48        painter.drawString("Distance:" + pointDistanceFormatted, x[i], y[i]);
49
50        // Display results
51        System.out.printf("Distance between point" + (i + 1) + " and line segment: %6.2f; " +
52                "Expected: %6.2f%n", toPointDistance[i], expectation[i]);
53    }
54    }
55 }
56
```

LineToPointDistance   DarkerDemo

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Distance between point1 and line segment: 70.71; Expected:  70.71
Distance between point2 and line segment:  0.00; Expected:   0.00
Distance between point3 and line segment: 141.42; Expected: 141.42
```
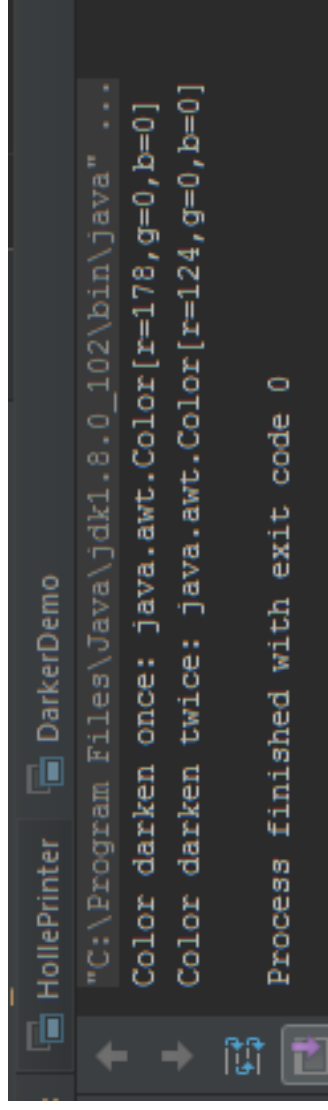
First Color

Second Co...

```
1  package Ch2Task.E2_10;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.10
9   * Darken two color objects once and twice
10  */
11 public class DarkerDemo {
12     public static void main(String[] args) {
13         // Construct first color object, darken it once
14         Color theFirstColor = Color.RED.darker();
15
16         // Construct first frame
17         JFrame firstFrame = new JFrame();
18         firstFrame.setSize(250, 250);
19         firstFrame.setTitle("First Color");
20         firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         firstFrame.setVisible(true);
22
23         // Apply first color to frame's background
24         firstFrame.getContentPane().setBackground(theFirstColor);
25
26         // Construct first color object, darken it twice
27         Color theSecondColor = theFirstColor.darker();
28
29         // Construct second frame
30         JFrame secondFrame = new JFrame();
31         secondFrame.setSize(250, 250);
```
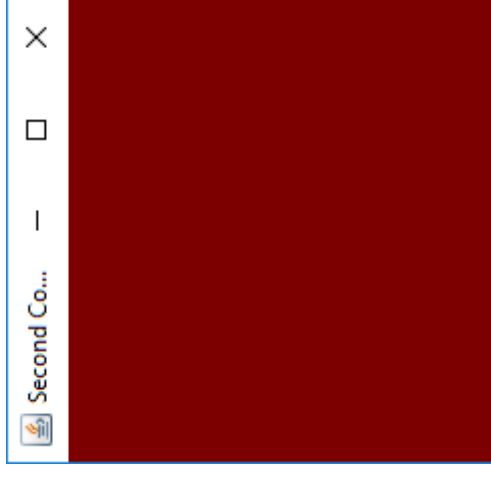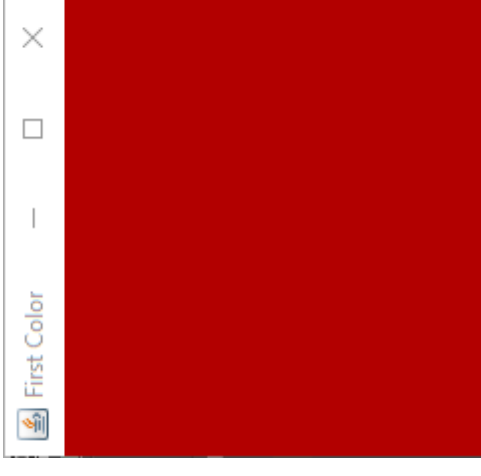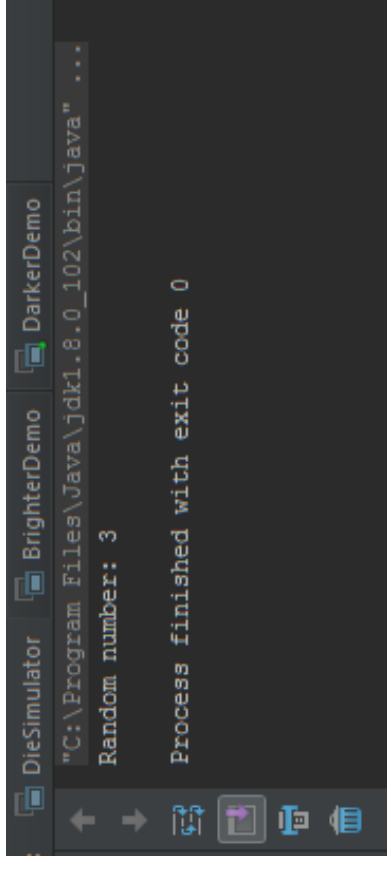
```java
32        secondFrame.setTitle("Second Color");
33        secondFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34        secondFrame.setVisible(true);
35
36        // Apply second color to frame's background
37        secondFrame.getContentPane().setBackground(theSecondColor);
38
39        // Display the result
40        System.out.println("Color darken once: " + theFirstColor);
41        System.out.println("Color darken twice: " + theSecondColor);
42    }
43  }
44
```
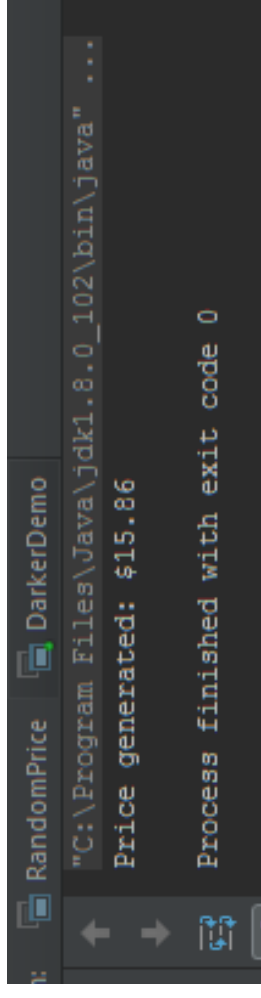
HollePrinter   DarkerDemo

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...
Color darken once: java.awt.Color[r=178,g=0,b=0]
Color darken twice: java.awt.Color[r=124,g=0,b=0]

Process finished with exit code 0
```

DieSimulator   BrighterDemo   DarkerDemo

```
"C:\Program Files\Java\jdk1.8.0_102\bin\java"...

Random number: 3

Process finished with exit code 0
```

```java
1  package Ch2Task.E2_11;
2
3  import java.util.Random;
4
5  /**
6   * Created by Sheldon on 10/23/2016.
7   * E2.11
8   * Generate random number
9   */
10 public class DieSimulator {
11     public static void main(String[] args) {
12         // Set the upper bound and lower bound to count
13         int max = 6;
14         int offset = 1;
15         int range = max - offset;
16
17         // Construct number generator object
18         Random generator = new Random();
19
20         // Generate random number
21         int inRangeGenerator = generator.nextInt(range) + offset;
22
23         // Display result
24         System.out.println("Random number: " + inRangeGenerator);
25     }
26 }
27
```

```java
1  package Ch2Task.E2_12;
2
3  import java.text.DecimalFormat;
4  import java.util.Random;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.12
9   * Generate random price from bound given
10  */
11 public class RandomPrice {
12     public static void main(String[] args) {
13         double maxPrice = 19.95;
14         double minPrice = 10.00;
15         double priceRange = maxPrice - minPrice;
16
17         Random generatePrice = new Random();
18         double inRangePrice = generatePrice.nextDouble() * priceRange + minPrice;
19
20         DecimalFormat rounder = new DecimalFormat(".##");
21         System.out.println("Price generated: $" + rounder.format(inRangePrice));
22     }
23 }
```

RandomPrice DarkerDemo

"C:\Program Files\Java\jdk1.8.0_102\bin\java" ...

Price generated: $15.86

Process finished with exit code 0

```java
1  package Ch2Task.P2_12;
2
3  import javax.swing.*;
4  import java.awt.*;
5  import java.awt.geom.Ellipse2D;
6
7  /**
8   * Created by Sheldon on 10/27/2016.
9   * P2.12
10  * Draws traffic light by 1 class
11  */
12 public class TrafficLight extends JComponent {
13     public static void main(String[] args) {
14         JFrame myFrame = new JFrame();
15         myFrame.setSize(700, 725);
16         myFrame.setTitle("Traffic Light");
17         myFrame.getContentPane().setBackground(Color.WHITE);
18         myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19         myFrame.setVisible(true);
20
21         TrafficLight theTrafficLight = new TrafficLight();
22         myFrame.add(theTrafficLight);
23     }
24     public void paintComponent(Graphics g) {
25         Graphics2D painter = (Graphics2D) g; // Construct painter object
26
27         // Draw traffic light's main body and cover
28         painter.setColor(Color.BLACK);
29         painter.fillRoundRect(350, 50, 150, 400, 20, 20);
30         painter.fillRoundRect(325, 38, 200, 15, 5, 5 );
31
```

```
32    // Draw white rectangular inside main body
33    painter.setColor(Color.WHITE);
34    painter.drawRoundRect(353, 53, 143, 393, 20, 20);
35
36    // Draw traffic light's stand
37    Rectangle trafficLightStand = new Rectangle(405, 450, 35, 200);
38    painter.setColor(Color.gray);
39    painter.fill(trafficLightStand);
40
41    // Draw the lamps
42    // RED lamp
43    Ellipse2D.Double redLamp = new Ellipse2D.Double(375, 75, 100, 100);
44    Color brightRed = Color.RED.brighter();
45    Color darkRed = Color.RED.darker();
46    painter.setColor(brightRed);
47    painter.fill(redLamp);
48    painter.setColor(Color.GRAY);
49    painter.draw(redLamp);
50    // Yellow lamp
51    Ellipse2D.Double yellowLamp = new Ellipse2D.Double(375, 200, 100, 100);
52    Color brightYellow = Color.YELLOW.brighter();
53    Color darkYellow = Color.YELLOW.darker();
54    painter.setColor(brightYellow);
55    painter.fill(yellowLamp);
56    painter.setColor(Color.GRAY);
57    painter.draw(yellowLamp);
58    // Green lamp
59    Ellipse2D.Double greenLamp = new Ellipse2D.Double(375, 325, 100, 100);
60    Color brightGreen = Color.GREEN.brighter();
61    Color darkGreen = Color.GREEN.darker();
62    painter.setColor(brightGreen);
```

```
63        painter.fill(greenLamp);
64        painter.setColor(Color.GRAY);
65        painter.draw(greenLamp);
66        // Draw lamps' cover
67        int arcWidth = 100;
68        int arcHeight = 100;
69        int arcStartAngle = 0;
70        int arcAngle = 180;
71        painter.setColor(Color.DARK_GRAY);
72        for (int i = 1; i < 30; i++) {
73            painter.drawArc(375, i + 73, arcWidth, arcHeight, arcStartAngle, arcAngle);
74            painter.drawArc(375, i + 198, arcWidth, arcHeight, arcStartAngle, arcAngle);
75            painter.drawArc(375, i + 323, arcWidth, arcHeight, arcStartAngle, arcAngle);
76        }
77    }
78 }
79
```