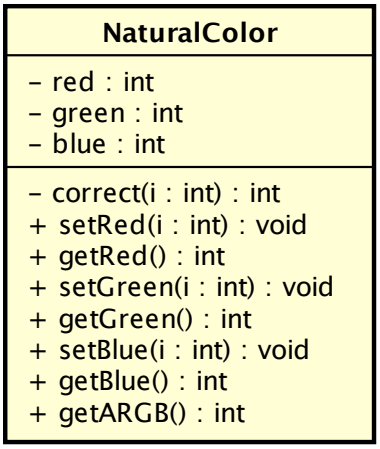
オブジェクト指向プログラミング 令和３年度 後期中間試験

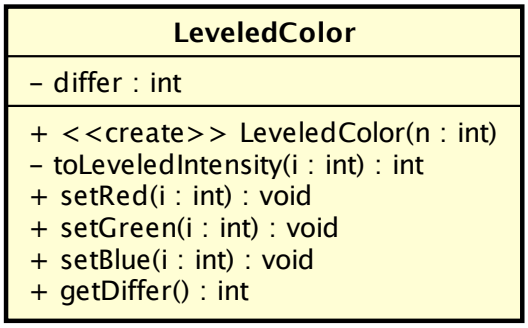
(2022.01.05 重村 哲至) IE5 _____番 氏名_____ **模範解答**

付録 1 にスクリーンショットとプログラムリストを掲載した Java アプリケーション (Prog1) について答えなさい．このアプリケーションは授業で取り上げた CurrentColor クラスを使ったアプリケーションを再設計したものです．なお，付録 2 にクラス図のサンプルを掲載するので参考にすること．

1. 以下に NaturalColor クラスのクラス図を描きなさい．クラス図はできる限り省略などをしないで正確に描くこと．(10 点)



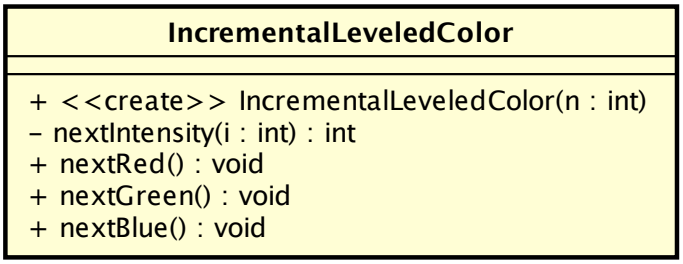
2. 以下に LeveledColor クラスのクラス図を描きなさい．クラス図はできる限り省略などをしないで正確に描くこと．なお，普通のメソッドとオーバーライドをするメソッドの表現方法の区別は UML のクラス図には無いので，普通のメソッドと同じ描き方をすればよい．(10 点)



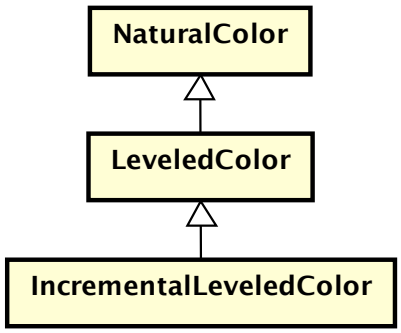
オブジェクト指向プログラミング 令和３年度 後期中間試験

(2022.01.05 重村 哲至) IE5 _____番 氏名_____ 模範解答

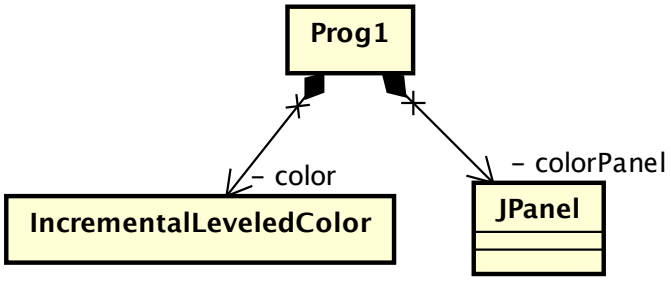
3. 以下に IncrementalLeveledColor クラスのクラス図を描きなさい．クラス図はできる限り省略などをしないで正確に描くこと．(10 点)



4. 以下に上記３つのクラスの関係を表現するクラス図を描きなさい．3 つのクラスの図は属性と操作を省略しクラス名だけの長方形で表現すること．(15 点)



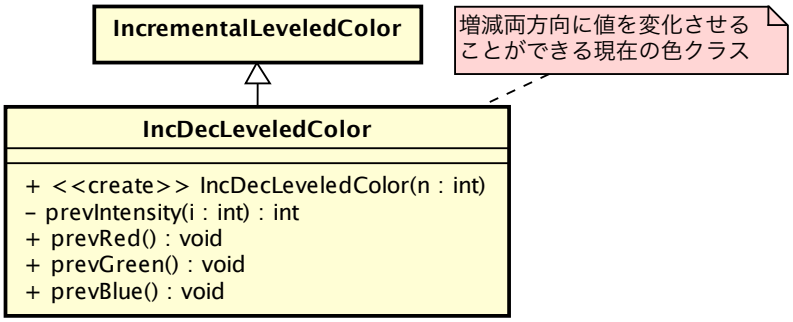
5. 以下に Prog1，IncrementalLevledColor，JPanel クラスの関連を表現するクラス図を描きなさい．3 つのクラスの図は属性と操作を省略しクラス名だけの長方形で表現すること．また，関連には，「誘導可能性 (矢印等のこと)」，「可視性 (+，-など) を含むロール名」，「集約など」を書き込むこと．(15 点)



オブジェクト指向プログラミング 令和３年度 後期中間試験

(2022.01.05 重村 哲至) IE5 ＿＿＿番 氏名＿＿＿＿＿＿ 模範解答

6. 次のクラス図は色成分の強さを増減両方向に変化させることができる「現在の色クラス」を表現しています．このクラス図に従って IncDecLeveledColor を実装した Java プログラムを書きなさい．なお，このクラスを実装するために，NaturalColor，LevledColor，IncrementalLeveledColor クラスに変更は加えないものとします．スーパークラスのメンバーの可視性に注意すること．(15 点)



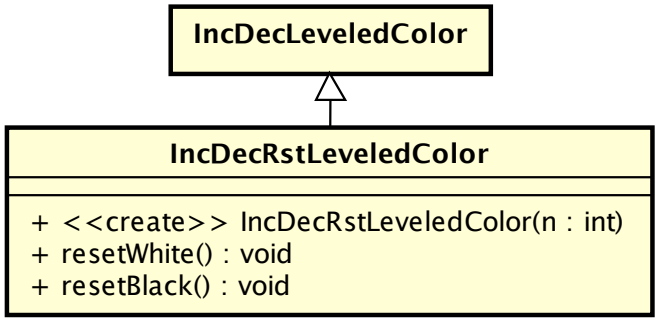

```
// 色成分を強くする方向，弱くする方向の両方に変化できる
public class IncDecLeveledColor
  extends IncrementalLeveledColor {
  public IncDecLeveledColor(int n) {
    super(n);
  }
  private int prevIntensity(int i) {
    i = i-getDiffer();
    if (i < 0) i = 255;
    return i;
  }
  public void prevRed() {
    setRed(prevIntensity(getRed()));
  }
  public void prevGreen() {
    setGreen(prevIntensity(getGreen()));
  }
  public void prevBlue() {
    setBlue(prevIntensity(getBlue()));
  }
}
```

オブジェクト指向プログラミング 令和３年度 後期中間試験

(2022.01.05 重村 哲至) IE5 _____番 氏名_____ **模範解答**

7. 色を白・黒にリセットする操作 `resetWhite()`, `resetBlack()` を持つ IncDecRstLeveledColor のクラス図を描き，Java プログラムの実装を書きなさい．次の点に注意すること．
   - IncDecLeveledColor クラスの代替として使用できること．
   - 他のクラスを継承する場合はスーパークラスの名前が分かるクラス図であること．
   - このクラスを実装するために，他のクラスの変更が不要なこと．

   (クラス図 10 点, Java プログラム 15 点)




```
// リセット機能を追加した
public class IncDecRstLeveledColor
  extends IncDecLeveledColor {
  public IncDecRstLeveledColor(int n) {
    super(n);
  }
  public void resetWhite() {
    setRed(255);
    setGreen(255);
    setBlue(255);
  }
  public void resetBlack() {
    setRed(0);
    setGreen(0);
    setBlue(0);
  }
}
```

付録１

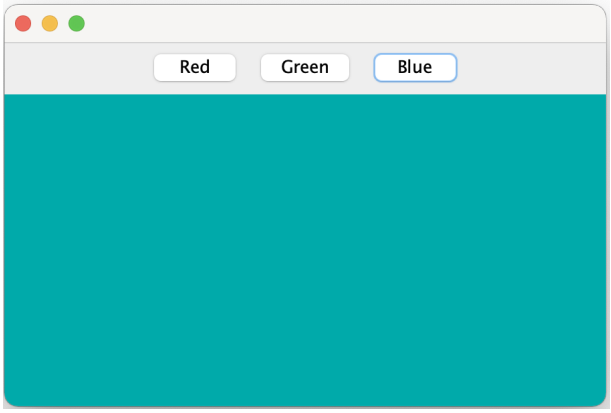


図 1-1: 実行例

リスト 1: NaturalColor クラス

```
1  // 普通の色を表現する自作のクラス
2  public class NaturalColor {
3    private int red;          // 赤成分の強度
4    private int green;        // 緑成分の強度
5    private int blue;         // 青成分の強度
6    // 不正な値を防ぐ
7    private int correct(int i) {
8      if (i<0) return 0;
9      if (i>255) return 255;
10     return i;
11   }
12   // アクセスメソッド
13   public void setRed(int i) {
14     red = correct(i);
15   }
16   public int getRed() {
17     return red;
18   }
19   public void setGreen(int i) {
20     green = correct(i);
21   }
22   public int getGreen() {
23     return green;
24   }
25   public void setBlue(int i) {
26     blue = correct(i);
27   }
28   public int getBlue() {
29     return blue;
30   }
31   public int getARGB() {
32     return (255<<24)|(red<<16)|(green<<8)|blue;
33   }
34 }
```

リスト 2: LeveledColor クラス

```
1  // RGB各成分に段階を設けて色を表現するクラス
2  public class LeveledColor extends NaturalColor {
3    private int differ;       // レベル間の強度の差
4    public LeveledColor(int n) {
5      differ = 256/n;
6    }
7    // レベル付きの強度へ変換
8    private int toLeveledIntensity(int i) {
9      return ((i + differ/2) / differ) * differ;
10   }
11   @Override
12   public void setRed(int i) {
13     // 親のsetRedを使用
14     super.setRed(toLeveledIntensity(i));
15   }
16   @Override
17   public void setGreen(int i) {
18     super.setGreen(toLeveledIntensity(i));
19   }
20   @Override
21   public void setBlue(int i) {
22     super.setBlue(toLeveledIntensity(i));
23   }
24   public int getDiffer() {
25     return differ;
26   }
27 }
```

リスト 3: IncrementalLeveledColor クラス

```
1  // 色成分を強くする方向に変化することができる
2  public class IncrementalLeveledColor extends LeveledColor {
3    public IncrementalLeveledColor(int n) {
4    // 親のコンストラクタを呼ぶ(引数付きのコンストラクタの場合は省略できない)
5      super(n);
6    }
7    private int nextIntensity(int i) {
8      i = i+getDiffer();
9      if (i > 255) i = 0;
10     return i;
11   }
12   public void nextRed() {
13     setRed(nextIntensity(getRed()));
14   }
15   public void nextGreen() {
16     setGreen(nextIntensity(getGreen()));
17   }
18   public void nextBlue() {
19     setBlue(nextIntensity(getBlue()));
20   }
21 }
```

リスト 4: Prog1 クラス

```
1  import java.awt.EventQueue;
2  ... 省略 ...
3  public class Prog1 {
4    private JFrame frame;
5    private IncrementalLeveledColor color = new IncrementalLeveledColor(3);
6    private JPanel colorPanel;
7
8    public static void main(String[] args) {
9      ... 省略 ...
10   }
11   public Prog1() {
12     initialize();
13   }
14   private void initialize() {
15     frame = new JFrame();
16     frame.setBounds(100, 100, 450, 300);
17     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18     colorPanel = new JPanel();
19     colorPanel.setBackground(new Color(color.getARGB()));
20     frame.getContentPane().add(colorPanel, BorderLayout.CENTER);
21     JPanel buttonPanel = new JPanel();
22     frame.getContentPane().add(buttonPanel, BorderLayout.NORTH);
23
24     JButton btnRed = new JButton("Red");
25     btnRed.addActionListener(new ActionListener() {
26       public void actionPerformed(ActionEvent e) {
27         color.nextRed();
28         colorPanel.setBackground(new Color(color.getARGB()));
29       }
30     });
31     buttonPanel.add(btnRed);
32
33     JButton btnGreen = new JButton("Green");
34     btnGreen.addActionListener(new ActionListener() {
35       public void actionPerformed(ActionEvent e) {
36         color.nextGreen();
37         colorPanel.setBackground(new Color(color.getARGB()));
38       }
39     });
40     buttonPanel.add(btnGreen);
41
42     JButton btnBlue = new JButton("Blue");
43     btnBlue.addActionListener(new ActionListener() {
44       public void actionPerformed(ActionEvent e) {
45         color.nextBlue();
46         colorPanel.setBackground(new Color(color.getARGB()));
47       }
48     });
49     buttonPanel.add(btnBlue);
50   }
51 }
```

付録２

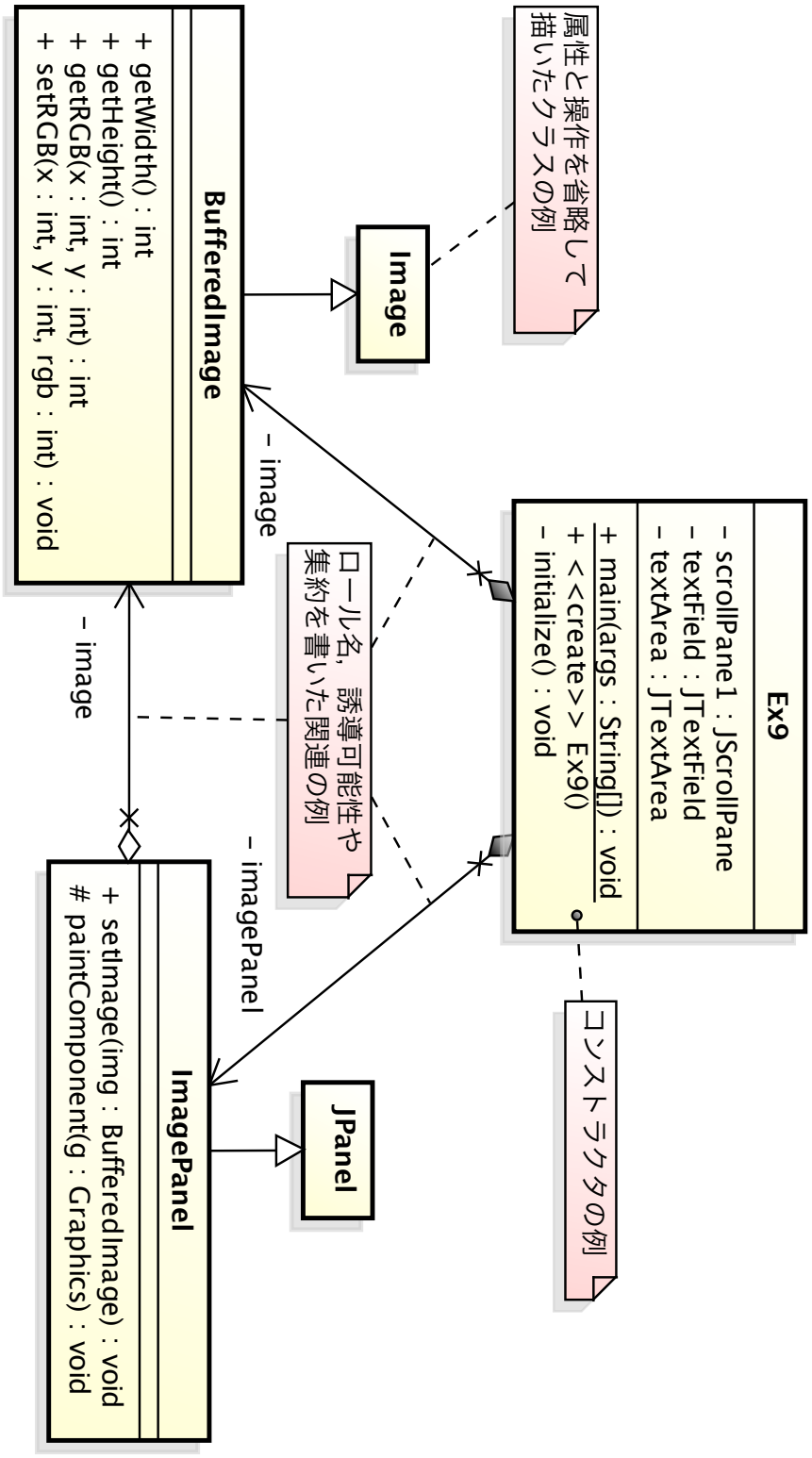


図 2-1: クラス図のサンプル