```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.stage.*;
import javafx.geometry.*;
import javafx.collections.*;

////////////////////////////////////////////////////////////
// 問１　下のように同じ画像（Food.jpg）を4つ表示してください…
////////////////////////////////////////////////////////////

public class Assignment5_1 extends Application
{
  public void start(Stage stage) throws Exception
  {
    // ４枚の画像を生成／設定します
    ImageView[] ivs = new ImageView[4];
    ivs[0] = new ImageView("Food.jpg");
    ivs[1] = new ImageView("Food.jpg");
    ivs[2] = new ImageView("Food.jpg");
    ivs[3] = new ImageView("Food.jpg");

    // エフェクトを生成し、画像に適用します
    GaussianBlur gb;
    gb = new GaussianBlur();
    gb.setRadius(5.0);
    ivs[1].setEffect(gb);
    gb = new GaussianBlur();
    gb.setRadius(10.0);
    ivs[2].setEffect(gb);
    gb = new GaussianBlur();
    gb.setRadius(20.0);
    ivs[3].setEffect(gb);

    // レイアウトHBoxを生成／設定します
    HBox hb = new HBox();
    ObservableList<Node> lst = hb.getChildren();
    lst.addAll(ivs);

    // シーンを生成／設定します
    Scene scene = new Scene(hb);

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("★ぼかしの効果★");

    // ステージを表示します
    stage.show();
  }

  public static void main(String[] args)
  {
    launch(args);
  }
}
```

```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.stage.*;
import javafx.geometry.*;
import javafx.collections.*;

////////////////////////////////////////////////////////////
// 問２　デジタル数字の画像ファイル（Digital.jpg）があります…
////////////////////////////////////////////////////////////

public class Assignment5_2 extends Application
{
  public void start(Stage stage) throws Exception
  {
    // ４枚の画像を生成／設定します
    ImageView[] ivs = new ImageView[4];
    ivs[0] = new ImageView("Digital.jpg");
    ivs[1] = new ImageView("Digital.jpg");
    ivs[2] = new ImageView("Digital.jpg");
    ivs[3] = new ImageView("Digital.jpg");

    // ビューポートを生成し、画像に適用します
    Rectangle2D[] rts = new Rectangle2D[4];
    rts[0] = new Rectangle2D(114*2, 183*0, 114, 183);
    rts[1] = new Rectangle2D(114*0, 183*0, 114, 183);
    rts[2] = new Rectangle2D(114*1, 183*0, 114, 183);
    rts[3] = new Rectangle2D(114*1, 183*1, 114, 183);
    for(int i=0;i<ivs.length;i++)
      ivs[i].setViewport(rts[i]);

    // レイアウトHBoxを生成／設定します
    HBox hb = new HBox();
    ObservableList<Node> lst = hb.getChildren();
    lst.addAll(ivs);

    // シーンを生成／設定します
    Scene scene = new Scene(hb);

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("★デジタル数字★");

    // ステージを表示します
    stage.show();
  }

  public static void main(String[] args)
  {
    launch(args);
  }
}
```

```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.stage.*;
import javafx.geometry.*;
import javafx.collections.*;

////////////////////////////////////////////////////////////
// 問３　下のように4枚の画像（Food.jpg Cafe.jpg Beach.jpg …
////////////////////////////////////////////////////////////

public class Assignment5_3 extends Application
{
  public void start(Stage stage) throws Exception
  {
    // ４枚の画像を準備します
    String[] files = {"Food.jpg","Cafe.jpg","Beach.jpg","Sunset.jpg"};
    Image[] imgs = new Image[files.length];
    for(int i=0;i<imgs.length;i++)
      imgs[i] = new Image(files[i]);

    // 画像表示枠の準備します
    ImageView[] ivs = new ImageView[3];
    for(int i=0;i<ivs.length;i++)
      ivs[i] = new ImageView();

    // 画像表示枠に初期の画像を設定します
    for(int i=0;i<ivs.length;i++)
      ivs[i].setImage(imgs[i%imgs.length]);

    // エフェクトを生成し、画像に適用します
    PerspectiveTransform ptleft = new PerspectiveTransform();
    ptleft.setUlx(160.0);
    ptleft.setUly(50.0);
    ptleft.setUrx(320.0);
    ptleft.setUry(0.0);
    ptleft.setLrx(320.0);
    ptleft.setLry(240.0);
    ptleft.setLlx(160.0);
    ptleft.setLly(190.0);
    PerspectiveTransform ptright = new PerspectiveTransform();
    ptright.setUlx(0.0);
    ptright.setUly(0.0);
    ptright.setUrx(160.0);
    ptright.setUry(50.0);
    ptright.setLrx(160.0);
    ptright.setLry(190.0);
    ptright.setLlx(0.0);
    ptright.setLly(240.0);

    // エフェクトを両側の画像表示枠に適用します
    ivs[0].setEffect(ptleft);
    ivs[2].setEffect(ptright);

    Thread th = new Thread(new SlideThread(imgs, ivs));
    th.start();

    // レイアウトHBoxを生成／設定します
    HBox hb = new HBox();
    ObservableList<Node> lst = hb.getChildren();
    lst.addAll(ivs);

    // シーンを生成／設定します
    Scene scene = new Scene(hb);
    scene.setFill(Color.BLACK);

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("★画像スライドショー★");

    // ステージを表示します
    stage.show();
  }

  public static void main(String[] args)
  {
    launch(args);
  }
}

// スライドスレッド
class SlideThread extends Thread
{
  // スライドを行う画像表示枠です
  private Image[] imgs;
  private ImageView[] ivs;

  // コンストラクタ
  public SlideThread(Image[] imgs, ImageView[] ivs)
  {
```

```
95      // スライドさせる画像表示枠を受け取ります
96      this.imgs = imgs;
97      this.ivs = ivs;
98    }
99
100   // １秒おきに画像をスライドします
101   public void run()
102   {
103     int count = ivs.length % imgs.length;
104     while(true)
105     {
106       try{
107         Thread.sleep(1000);
108       }catch(InterruptedException e){}
109
110       ivs[0].setImage(ivs[1].getImage());
111       ivs[1].setImage(ivs[2].getImage());
112       ivs[2].setImage(imgs[count]);
113       count++;
114       count %= imgs.length;
115     }
116   }
117 }
118
```