
Data Structures and Algorithms

COMP 103

2019-20

Semester 2

Lecture 02b

Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

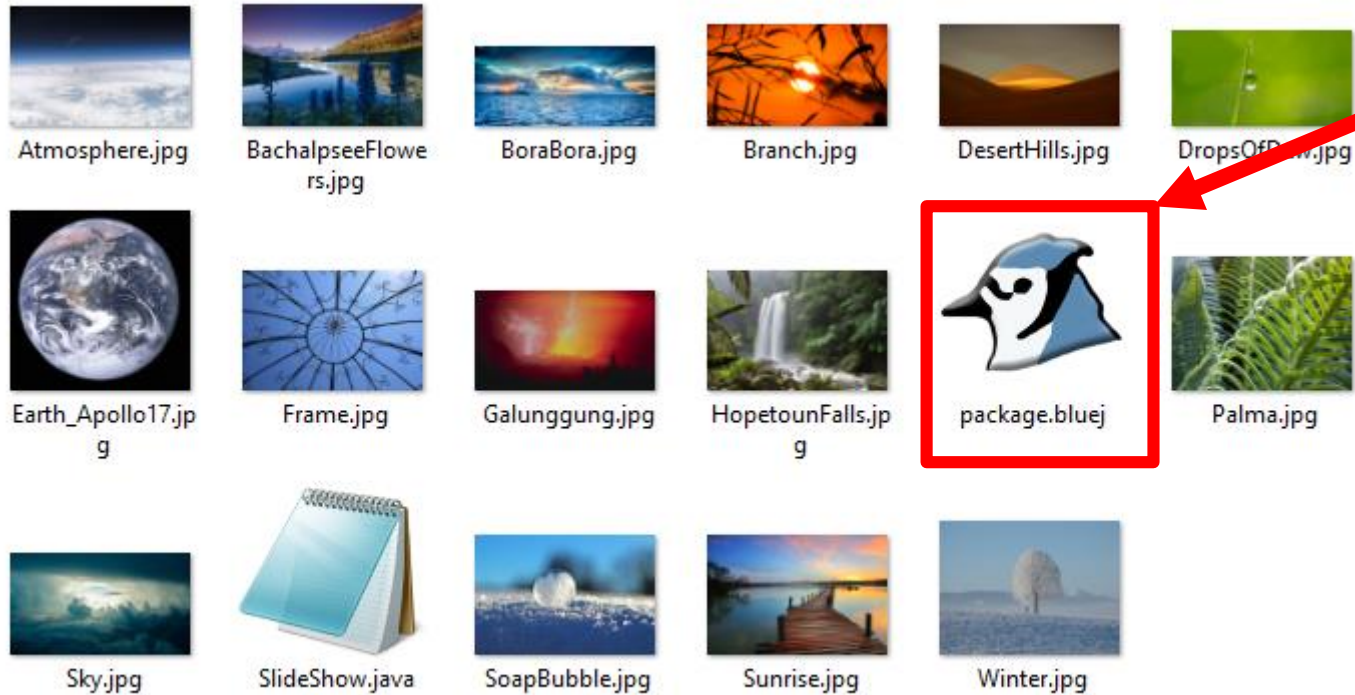
Victoria University of Wellington

Download & unzip the file

Name	Date modified	Type	Size
SlideShow	1/04/2020 9:23 AM	File folder	
XMUT103-2020T1-Assig1PartA.zip	1/04/2020 9:18 AM	WinZip File	1,031 KB

Double-click on the package.bluej file to start the BlueJ program & open the SlideShow.java file

SlideShow folder files



Class Edit Tools Options

SlideShow ×

Compile Undo Cut Copy Paste Find... Close

Source Code ▾

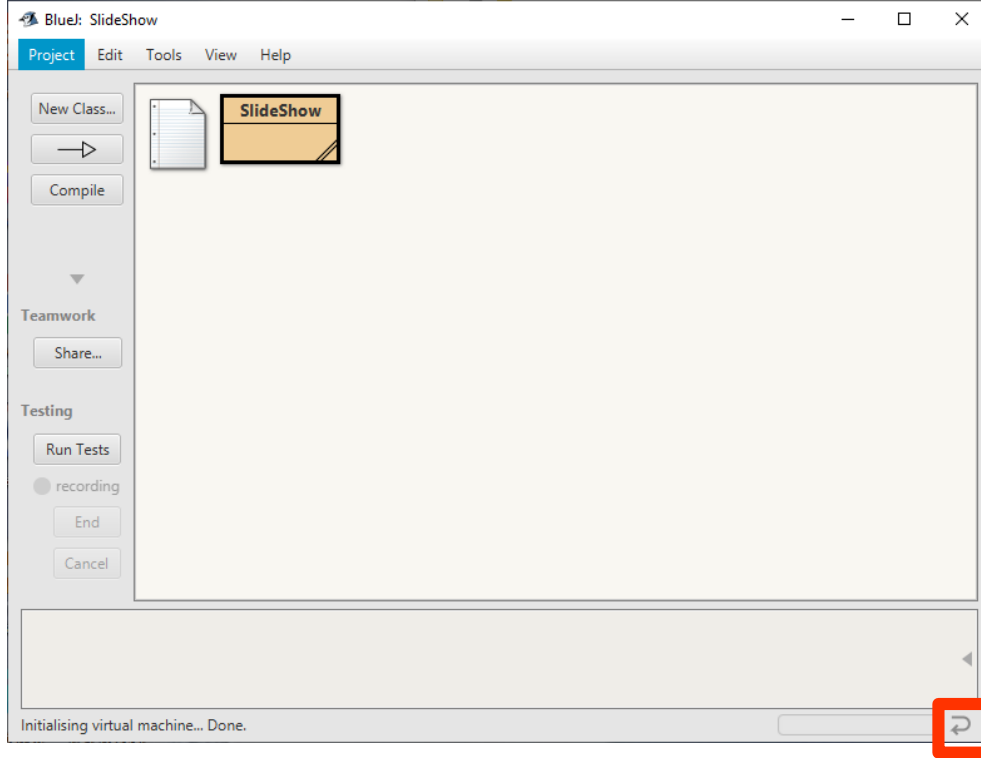
```
122 public void dokey(String key) {
123     /*
124         if (key.equals("Left"))        goLeft();
125         else if (key.equals("Right"))   goRight();
126         else if (key.equals("Home"))    goStart();
127         else if (key.equals("End"))     goEnd(); */
127     }
128
129
130 /**
131  * A method that adds a bunch of names to the list of images, for testing.
132  */
133 public void setTestList(){
134     if (showRunning) return;
135     String[] names = new String[] {"Atmosphere.jpg", "BachalpseeFlowers.jpg",
136         "BoraBora.jpg", "Branch.jpg", "DesertHills.jpg",
137         "DropsOfDew.jpg", "Earth_Apollo17.jpg",
138         "Frame.jpg", "Galunggung.jpg", "HopetounFalls.jpg",
```



Click on Compile button

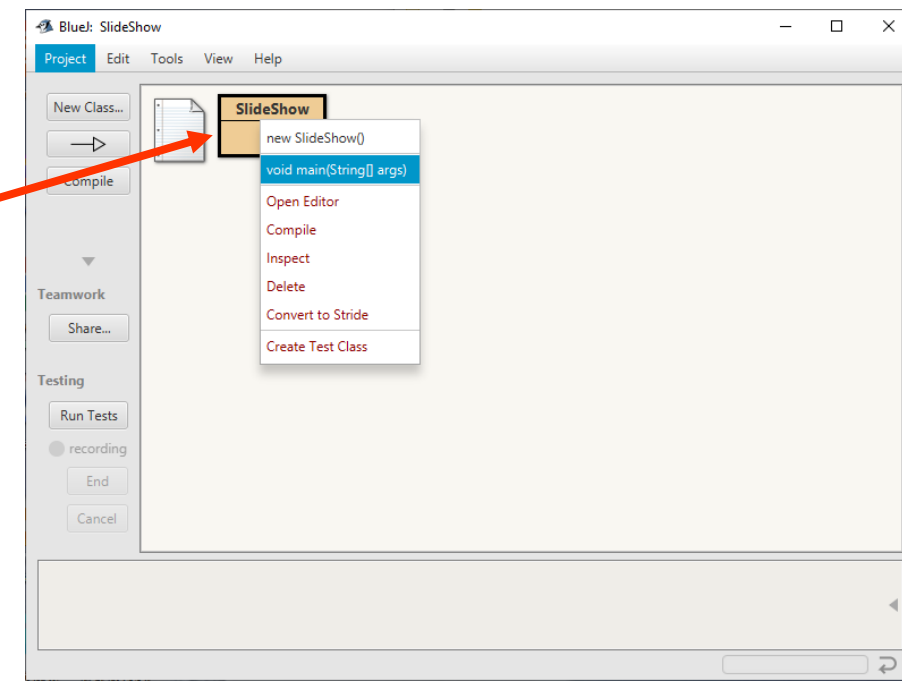
Class compiled - no syntax errors

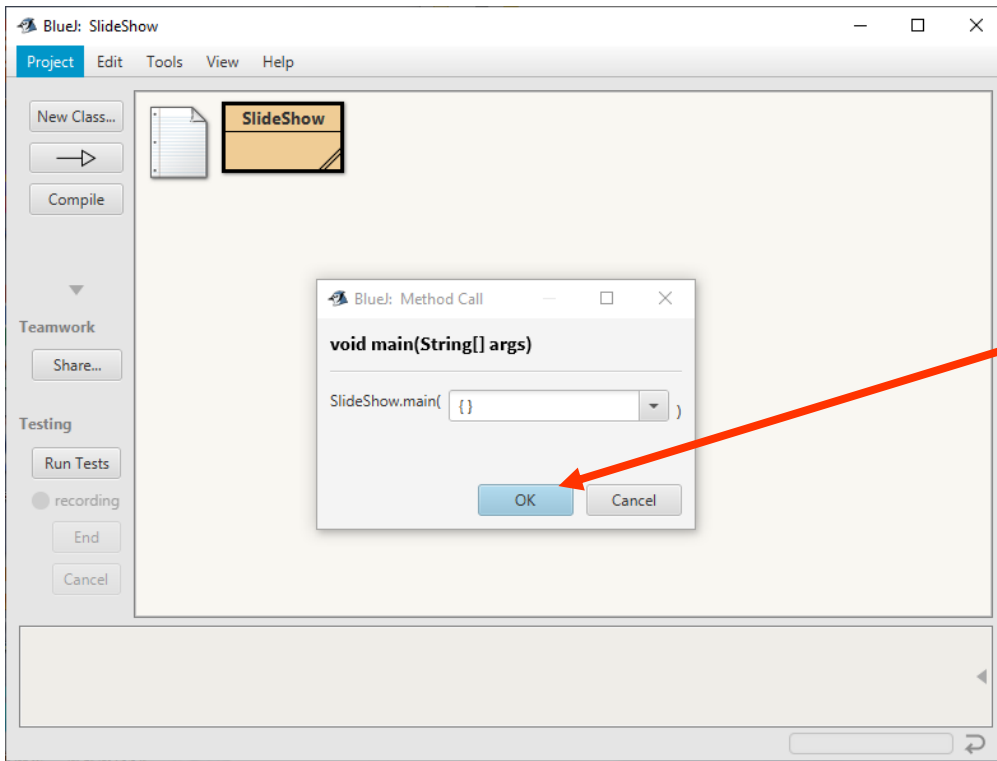
saved



1. Click on this Reset icon to initialize/reset the Java Virtual Machine

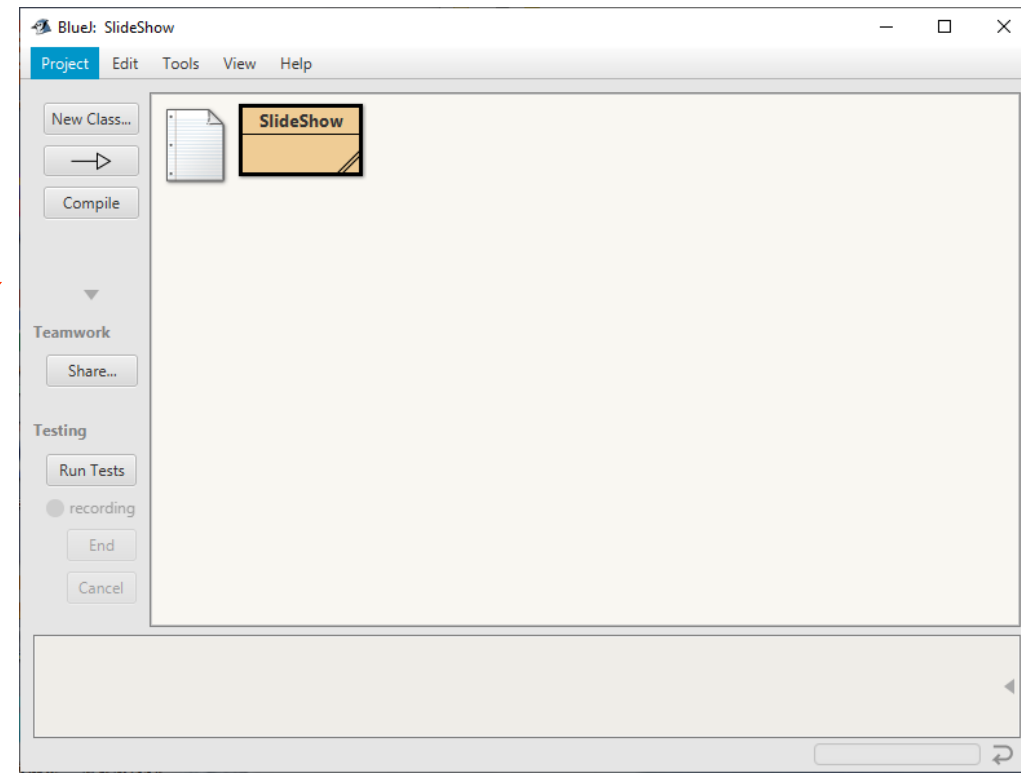
2. Click the right-hand-side button on your mouse on the SlideShow rectangle and select the second option as shown on the screenshot





1. Click on the OK button to run the SlideShow program

2. Program has run successfully but no visual output is produced!



```
1 // This program is copyright VUW.  
2 // You are granted permission to use it to construct your answer to a COMP103 assignment.  
3 // You may not distribute it in any other way without permission.  
4
```

```
5 /* Code for COMP103 - 2020T1, Assignment 1  
6 * Name: Kerese Manueli  
7 * Username: XMUT20201213456  
8 * ID: 20201213456  
9 */
```

```
11 import java.util.*;  
12 import ecs100.*;  
13 import java.awt.Color;
```

← 3 import statements

```
15 /**  
16 * This class contains the main method of the program.  
17 *
```

```
18 * A SlideShow object represents the slideshow application and sets up the buttons in the UI.
```

```
19 *
```

```
20 * @author pondy
```

```
21 */
```

```
22 public class SlideShow {
```

```
24     public static final int LARGE_SIZE = 450;    // size of images during slide show
```

```
25     public static final int SMALL_SIZE = 100;   // size of images when editing list
```

```
26     public static final int GAP = 10;          // gap between images when editing
```

```
27     public static final int COLUMNS = 6;      // Number of columns of thumbnails
```

```
30     private List<String> images; // List of image file names.
```

```
32     private int currentImage = -1; // index of currently selected image.
```

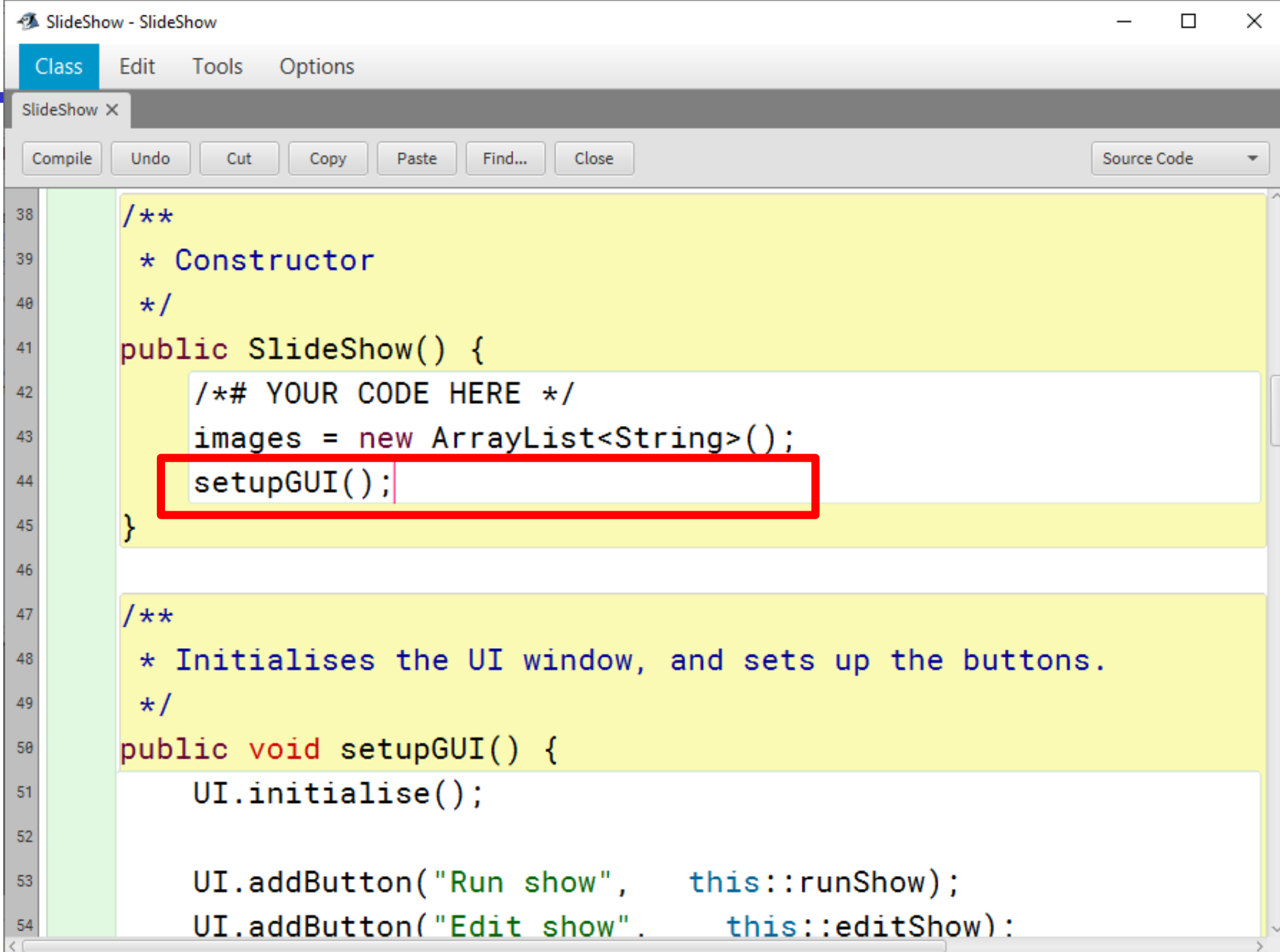
```
33     // Should always be a valid index if there are any images
```

```
130 /**
131  * A method that adds a bunch of names to the list of images, for testing.
132  */
133 public void setTestList(){
134     if (showRunning) return;
135     String[] names = new String[] {"Atmosphere.jpg", "BachalpseeFlowers.jpg",
136     "BoraBora.jpg", "Branch.jpg", "DesertHills.jpg",
137     "DropsOfDew.jpg", "Earth_Apollo17.jpg",
138     "Frame.jpg", "Galunggung.jpg", "HopetounFalls.jpg",
139     "Palma.jpg", "Sky.jpg", "SoapBubble.jpg",
140     "Sunrise.jpg", "Winter.jpg"};
141
142     for(String name : names){
143         images.add(name);
144     }
145     currentImage = 0;
146     display();
```



```
32 private int currentImage = -1;    // index of currently selected image.
33 // Should always be a valid index if there are any images
34
35 private boolean showRunning;    // flag signalling whether the slideshow is running or not
36
37
38 /**
39  * Constructor
40  */
41 public SlideShow() {
42     /*# YOUR CODE HERE */
43 }
44
45 /**
46  * Initialises the UI window, and sets up the buttons.
47  */
48 public void setupGUI() {
49     UI initialisation();
50 }
```

```
29
30 private List<String> images; // List of image file names.
31
32 private int currentIndex = -1; // index of currently selected image.
33 // Should always be a valid index if there are any images
34
35 private boolean showRunning; // flag signalling whether the slideshow is running or not
36
37
38 /**
39  * Constructor
40  */
41 public SlideShow() {
42     /*# YOUR CODE HERE */
43     images = new ArrayList<String>();
44 }
45
```



```
38  /**
39  * Constructor
40  */
41  public SlideShow() {
42      /*# YOUR CODE HERE */
43      images = new ArrayList<String>();
44      setupGUI();
45  }
46
47  /**
48  * Initialises the UI window, and sets up the buttons.
49  */
50  public void setupGUI() {
51      UI.initialise();
52
53      UI.addButton("Run show",    this::runShow);
54      UI.addButton("Edit show",   this::editShow);
```