1 **import** java.text.DecimalFormat;

2 **import** javafx.application.Application;

3 **import** javafx.event.\*;

4 **import** javafx.geometry.\*;

5 **import** javafx.scene.Scene;

6 **import** javafx.scene.control.\*;

7 **import** javafx.scene.layout.\*;

8 **import** javafx.scene.text.Font;

9 **import** javafx.stage.Stage;

10

11 **public** **class** AddingMachineV2 **extends** Application

12 {

13 Label description, plus, equals, sum, a, b;

14 TextField aValue, bValue;

15 Button compute, clear;

16

17 @Override

18 **public** **void** start(Stage primaryStage)

19 {

20 //Step1: Declare the component objects, Step2: Set their properties

21 description = **new** Label("Computes a + b");

22 description.setFont(new Font("Arial", 24));

23 aValue = **new** TextField();

24 aValue.setPrefSize(120, 30); plus = new Label("+");

25 plus.setFont(**new** Font("Arial", 24));

26 bValue = **new** TextField();

27 bValue.setPrefSize(120, 30); equals = new Label("=");

28 equals.setFont(**new** Font("Arial", 24));

29 sum = **new** Label("x,xxx.xx");

30 sum.setFont(**new** Font("Arial", 16));

31 sum.setMinWidth(68);

32 a = **new** Label("a");

33 a.setFont(new Font("Arial", 24));

34 b = **new** Label("b");

35 b.setFont(**new** Font("Arial", 24));

36 compute = **new** Button(" Compute ");

37 compute.setOnAction( (e) -> computeClickHandler() );

38 clear = **new** Button (" Clear ");

39 clear.setOnAction( (e) -> clearClickHandler() );

40 Tooltip.install(clear, new Tooltip("Clears operands and result"));

41

42 //Step 3: Declare the component container, and set its properties

43 GridPane grid = new GridPane();

44 grid.setHgap(10);

45 grid.setVgap(10);

46 grid.setPadding(new Insets(10, 0, 10, 10));

47 grid.setHalignment(description, HPos.CENTER);

48 grid.setHalignment(a, HPos.CENTER);

49 grid.setHalignment(b, HPos.CENTER);

50 grid.setHalignment(compute, HPos.CENTER);

51 grid.setHalignment(clear, HPos.CENTER);

52

53 //Step 4: Add the components to the component container

54 grid.add(description, 0, 0, 5, 1);

55 grid.add(aValue, 0, 1, 1, 1);

56 grid.add(plus, 1, 1, 1, 1);

57 grid.add(bValue, 2, 1, 1, 1);

58 grid.add(equals, 3, 1, 1, 1);

59 grid.add(sum, 4, 1, 1, 1);

60 grid.add(a, 0, 2, 1, 1);

61 grid.add(b, 2, 2, 1, 1);

62 grid.add(compute, 0, 3, 1, 1);

63 grid.add(clear, 2, 3, 1, 1);

64

65

66 Scene scene = **new** Scene(grid); // add the container to the scene

67

68 primaryStage.setTitle("Calculator");

69 primaryStage.setScene(scene);

70 primaryStage.show();

71 }

72

73 **private** **void** computeClickHandler()

74 {

75 String s;

76 double a, b, result;

77 DecimalFormat f = **new** DecimalFormat("#,##0.00");

78

79 s = aValue.getText();

80 a = Double.parseDouble(s);

81 s = bValue.getText();

82 b = Double.parseDouble(s);

83 result = a + b;

84 sum.setText(f.format(result));

85 }

86

87 **private** **void** clearClickHandler()

88 {

90 aValue.setText("");

91 bValue.setText("");

92 sum.setText("x,xxx.xx");

93 }

94

96 **public** **static** **void** main(String[] args)

97 {

98 launch(args);

99 }

100 }

**Figure 11.19**

The application AddingMachineV2