JavaFX Projects

HECTOR MATHONSI

Enhance Your Java Skills with Fun Projects

"***The only way to learn a new programming language is by writing programs in it.***" - Dennis Ritchie

Contents

Introduction

Welcome to your JavaFX Projects for Winter Recess! This document is designed to provide you with engaging JavaFX programming challenges to sharpen your Java skills while enjoying your winter break. Each project is crafted to reinforce key concepts and encourage creative problem-solving.

About JavaFX

JavaFX is a powerful framework for building interactive applications in Java. It provides a rich set of tools and libraries for creating modern user interfaces, animations, and multimedia experiences.

How to Use This Document

1. Each project presents a unique problem statement and task.
2. Read through the problem description carefully.
3. Apply your Java skills to implement the solution.
4. Experiment with different features of JavaFX to enhance your projects.
5. Have fun and enjoy the learning process!

Recommended Prerequisites

- Basic understanding of Java programming language.
- Familiarity with object-oriented programming concepts.
- Knowledge of event-driven programming.

Happy Coding!

Enjoy tackling these JavaFX projects and explore the world of Java development. Use this opportunity to expand your programming skills and unleash your creativity.

JavaFX Basics

Topics Covered: Panes, Groups, UI controls, Shapes, Property Binding, Style and Rotate Properties, Color class, Font class, Image class, ImageView class, Layout Nodes, Text class, Shape class, ClockPane.

Problem 1: Display images

Write a program that displays four images in a grid pane, as shown in the following Figure. You can use any pictures for this problem.
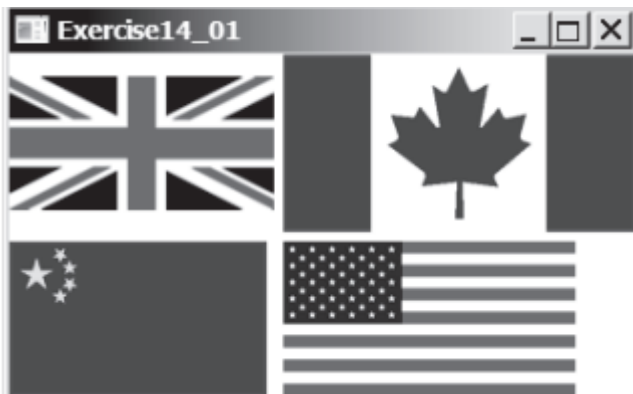
Problem 2: Color and Font

Write a program that displays five texts vertically, as shown in the following Figure. Set a random color and opacity for each text and set the font of each text to Times Roman, bold, italic, and 24 pixels.

Problem 3: Display random 0 or 1

Write a program that displays a 10-by-10 square matrix, as shown in the following Figure. Each element in the matrix is 0 or 1, randomly generated. Display each number centered in a text field. Use **TextField**'s setText method to set value 0 or 1 as a string.

0	0	1	0	0	1	1	0	0	0
1	1	0	1	0	0	0	0	1	0
1	1	0	0	0	0	1	0	0	1
1	0	1	0	0	0	1	1	0	0
1	0	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	0	1
1	0	1	0	1	1	0	0	0	1
1	1	0	1	0	0	0	1	0	1
0	1	1	0	1	0	1	0	0	0

Problem 4: ClockPane class & Random time

a. Write a program that displays two clocks. The hour, minute, and second values are 4, 20, 45 for the first clock, and 22, 46, 15 for the second clock, as shown in the following Figure.
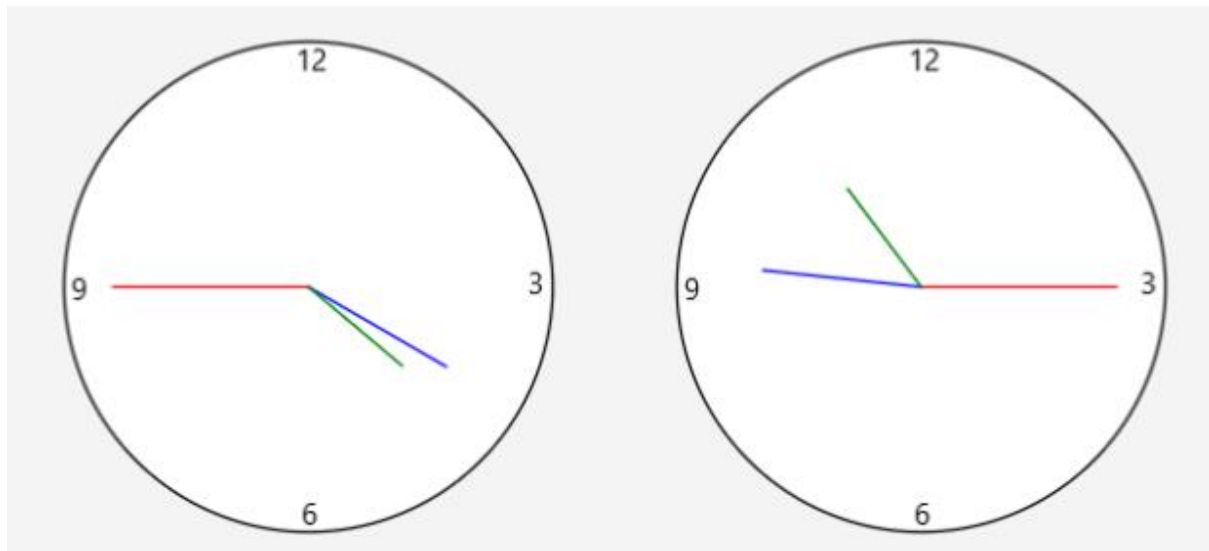
b. Modify the ClockPane class with three new Boolean properties **hourHandVisible**, **minuteHandVisible**, and **secondHandVisible** and their associated accessor and mutator methods. You can use the set methods to make a hand visible or invisible. Write a test program that displays only the hour and minute hands. The hour and minute values are randomly generated. The hour is between 0 and 11, and the minute is either 0 or 30, as shown in the following Figure.
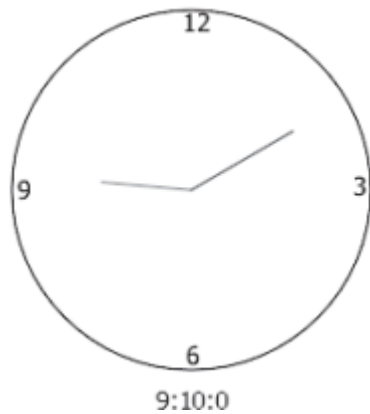

Event-Driven Programming and Animations

Topics Covered: JavaFX Basics, Event-driven Programming, Events, Event sources, Event classes, Handler classes, Inner classes, Anonymous Inner classes, Lambda Expressions, **MouseEvent**s, **KeyEvent**s, Listeners for Observable Objects, Animation.

Problem 1: Pick four cards

Write a program that lets the user click the Refresh button to display four cards from a deck of 52 cards, as shown in the following Figure.
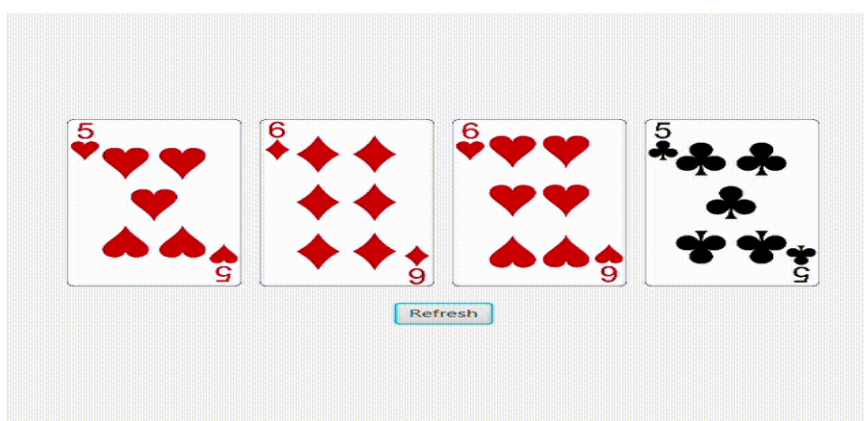
Problem 2: Move the rectangle

Write a program that moves the rectangle in a pane. You should define a pane class for displaying the ball and provide the methods for moving the Rectangle left, right, up, and down,

as shown in the following Figure. Check the boundary to prevent the ball from moving out of sight completely.
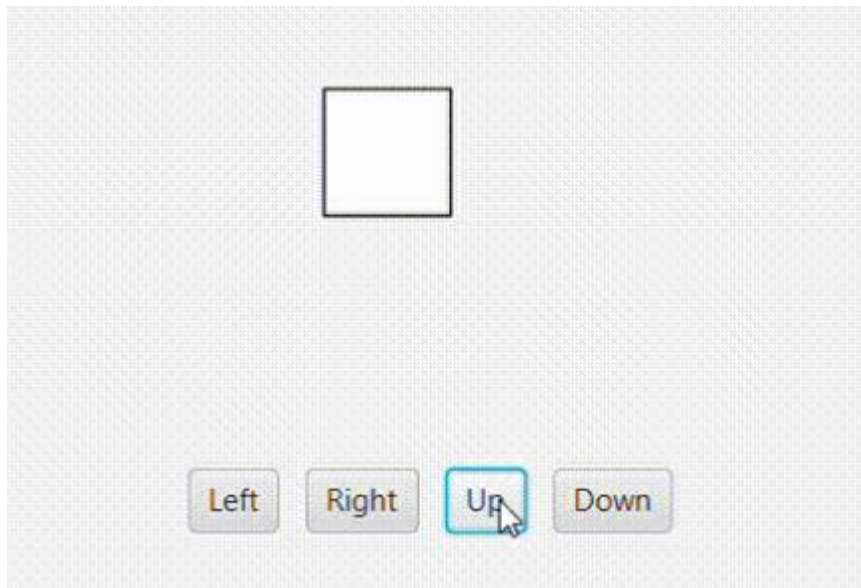

Problem 3: Draw lines using the arrow keys

Write a program that draws line segments using the arrow keys. The line starts from (100, 100) in the pane and draws toward east, north, west, or south when the right-arrow key, up-arrow key, left-arrow key, or down-arrow key is pressed, as shown in the following Figure.


Problem 4: Geometry: inside a circle?

Write a program that draws a fixed circle centered at (100, 60) with radius 50. Whenever the mouse is moved, display a message indicating whether the mouse point is inside the circle at the mouse point or outside of it, as shown in the following Figure.

point (88.80, 47.20) inside the circle
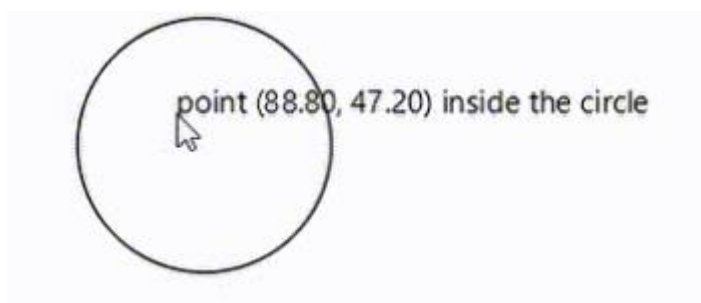
Problem 5: Geometry: find the bounding rectangle

Write a program that enables the user to add and remove points in a two-dimensional plane dynamically, as shown in the following Figure. A minimum bounding rectangle is updated as the points are added and removed. Assume the radius of each point is 10 pixels.
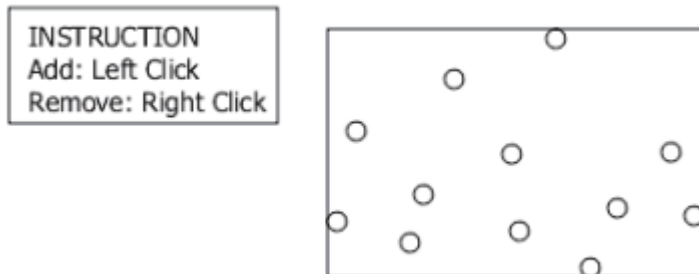
Problem 6: Simulation: self-avoiding random walk

A self-avoiding walk in a lattice is a path from one point to another that does not visit the same point twice. Self-avoiding walks have applications in physics, chemistry, and mathematics. They can be used to model chain-like entities such as solvents and polymers. Write a program that displays a random path that starts from the center and ends at a point on the boundary, as shown in the following Figure, or ends at a dead-end point (i.e., surrounded by four points that have already been visited), as shown in the following Figure. Assume the size of the lattice is 16 by 16.
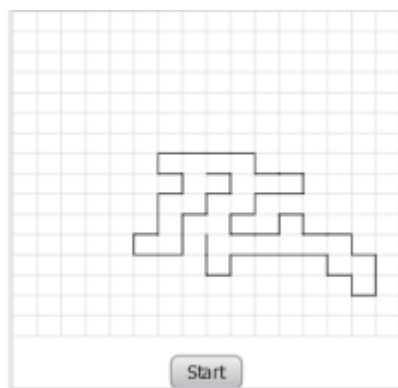
JavaFX UI Controls and Multimedia

Topics Covered: JavaFX Basics, **Event-Driven Programming and Animations**, Labeled and Label, Button, CheckBox, RadioButton, TextField, TextArea, ComboBox, ListView, ScrollBar, Slider.

Problem 1: Use radio buttons

Write a GUI program as shown in the following Figure. You can use buttons to move the message to the left and right and use the radio buttons to change the color for the message displayed.
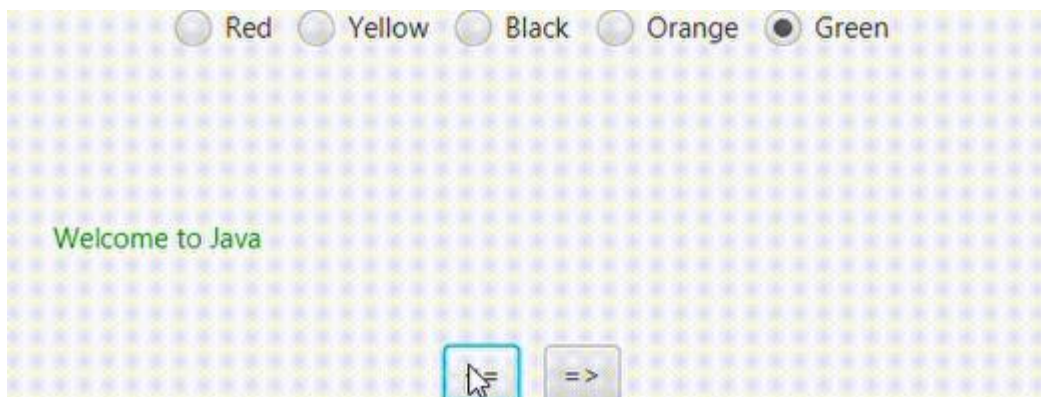
Problem 2: Select geometric figures

Write a program that draws various figures, as shown in the following Figure. The user selects a figure from a radio button and uses a check box to specify whether it is filled.
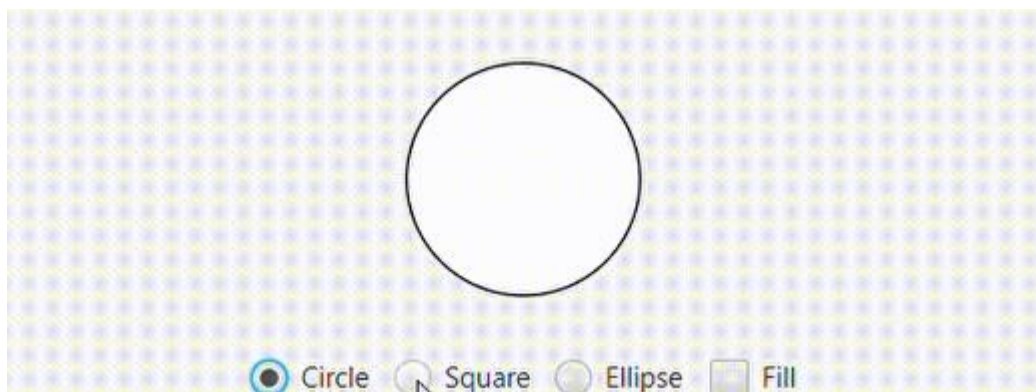
Problem 3: Select a font

Write a program that can dynamically change the font of a text in a label displayed on a stack pane. The text can be displayed in bold and italic at the same time. You can select the font name or font size from combo boxes, as shown in the following Figure. The available font names can be obtained using Font.getFontNames(). The combo box for the font size is initialized with numbers from 1 to 100.

Problem 4: Pattern recognition: consecutive four equal numbers

Write a GUI program as shown in the following Figures. Let the user enter the numbers in the text fields in a grid of 6 rows and 7 columns. The user can click the Solve button to highlight a sequence of four equal numbers, if it exists. Initially, the values in the text fields are randomly filled with numbers from 0 to 9.
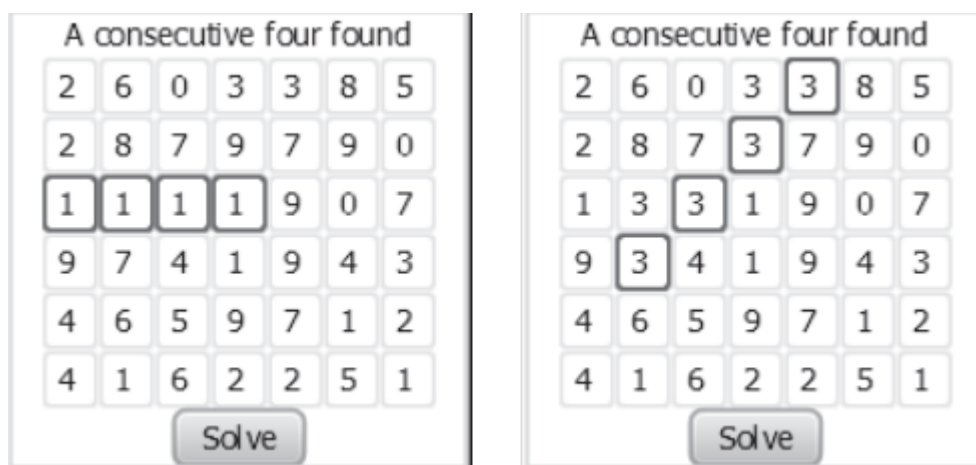
Bibliography

Liang, Y. D. (2018). *Introduction to Java Programming and Data Structures* (11th ed.). United Kingdom: Pearson Education.