Design Overview for Option 2.

Name: Dave Nguyen.
Student ID: 104697710.

Summary of Program

This sketching program is an interactive drawing tool that lets users draw and modify shape objects. Users may use keyboard keys (R, C, L, E) to pick a shape type and then click on the canvas to create rectangles, circles, lines, and ellipses. The software has basic editing features such as right-clicking to pick shapes, removing selected forms, changing the background's color, and saving/loading drawings to save progress between sessions.

Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: <<role name>> details – duplicate

Responsibility	Type Details	Notes
Main entry point and game loop management	Fields: ShapeKind shapeToAdd<>Methods: static void Main()	Contains the main loop that processes user input, handles keyboard and mouse events, and orchestrates interactions between the user and drawing object.
Keyboard input handling	Parameters: KeyCode enums<>Return: void	Detects key presses (R, C, L, E, D, G, Q, M, Space, Backspace, Delete, S, O) and triggers appropriate actions.
Mouse input handling	Parameters: MouseButton, Point2D<>Return: void	Processes left clicks for shape creation and right clicks for shape selection.
Manages the collection of shapes	Fields: List<Shape> _shapes, Color _background, Random _random<>Return: void	Central container that maintains all shapes on the canvas.
Shape creation and manipulation	Methods: AddShape(Shape s), RemoveShape(Shape s), RemoveLastShape()<>Parameters: Shape objects<>Return: void	Provides methods to add, remove, and manage shapes in the collection.

Rendering all shapes	Method: Draw()<>Return: void	Iterates through all shapes and calls their Draw() methods, handles background clearing.
Random shape generation	Method: GenerateRandomShapes()<>Return: void	Creates 5-14 random shapes with random positions, colors, and dimensions.
Color randomization	Method: RandomizeAllColors()<>Return: void	Changes the colors of all existing shapes to random values.
Text drawing	Methods: DrawDAVE(), DrawLetterD(float, float), DrawLetterA(float, float), DrawLetterV(float, float), DrawLetterE(float, float)<>Parameters: float x, float y<>Return: void	Draws the first name "DAVE" using shapes; each letter method creates specific shape combinations that resemble the letter of my name.
Shape scaling	Method: ScaleDownAllShapes()<>Return: void	Calls ScaleDown() on each shape in the collection.
Persistence	Methods: Save(string filename), Load(string filename)<>Parameters: string<>Return: void	Saves and loads drawings to/from text files.
Shape selection	Method: SelectedShapesAt(Point2D pt)<>Parameters: Point2D<>Return: void	Marks shapes at the given point as selected.
Property management	Properties: SelectedShapes (List<Shape>), ShapeCount (int), Background (Color)<>Return: List<Shape>, int, Color	Provides access to selected shapes, total count, and background color.
Define common shape properties.	Fields: Color _color, float _x, float _y, bool _selected<>Return: various	Base class providing shared attributes for all shape types.
Property accessors	Properties: Color, X, Y, Selected<>Return: Color, float, float, bool	Get/set methods for shape properties.
Abstract drawing contract	Method: abstract void Draw()<>Return: void	Forces all derived classes to implement their own drawing logic.
Abstract outline drawing	Method: abstract void DrawOutline()<>Return: void	Forces all derived classes to implement the selection outline.
Abstract hit detection	Method: abstract bool IsAt(Point2D pt)<>Parameters: Point2D<>Return: bool	Forces all derived classes to implement point containment checking.

Abstract scaling contract	Method: abstract void ScaleDown()<>Return: void	Forces all derived classes to implement size reduction.
Persistence support	Methods: virtual void SaveTo(StreamWriter), virtual void LoadFrom(StreamReader)<>Parameters: StreamWriter/StreamReader<>Return: void	Saves/loads common properties.
Rectangle scaling	Method: override void ScaleDown()<>Return: void	Reduces width and height.
Circle scaling	Method: override void ScaleDown()<>Return: void	Reduces width and height.
Line scaling	Method: override void ScaleDown()<>Return: void	Reduces width and height.
Ellipse scaling	Method: override void ScaleDown()<>Return: void	Reduces width and height.

Table 2: <<enumeration name>> details

Value	**Notes**
Rectangle	Represents a rectangle shape type for user selection.
Circle	Represents circle shape type for user selection.
Line	Represents line shape type for user selection.
Ellipse	Represents the ellipse shape type for user selection.

Sequence Diagram

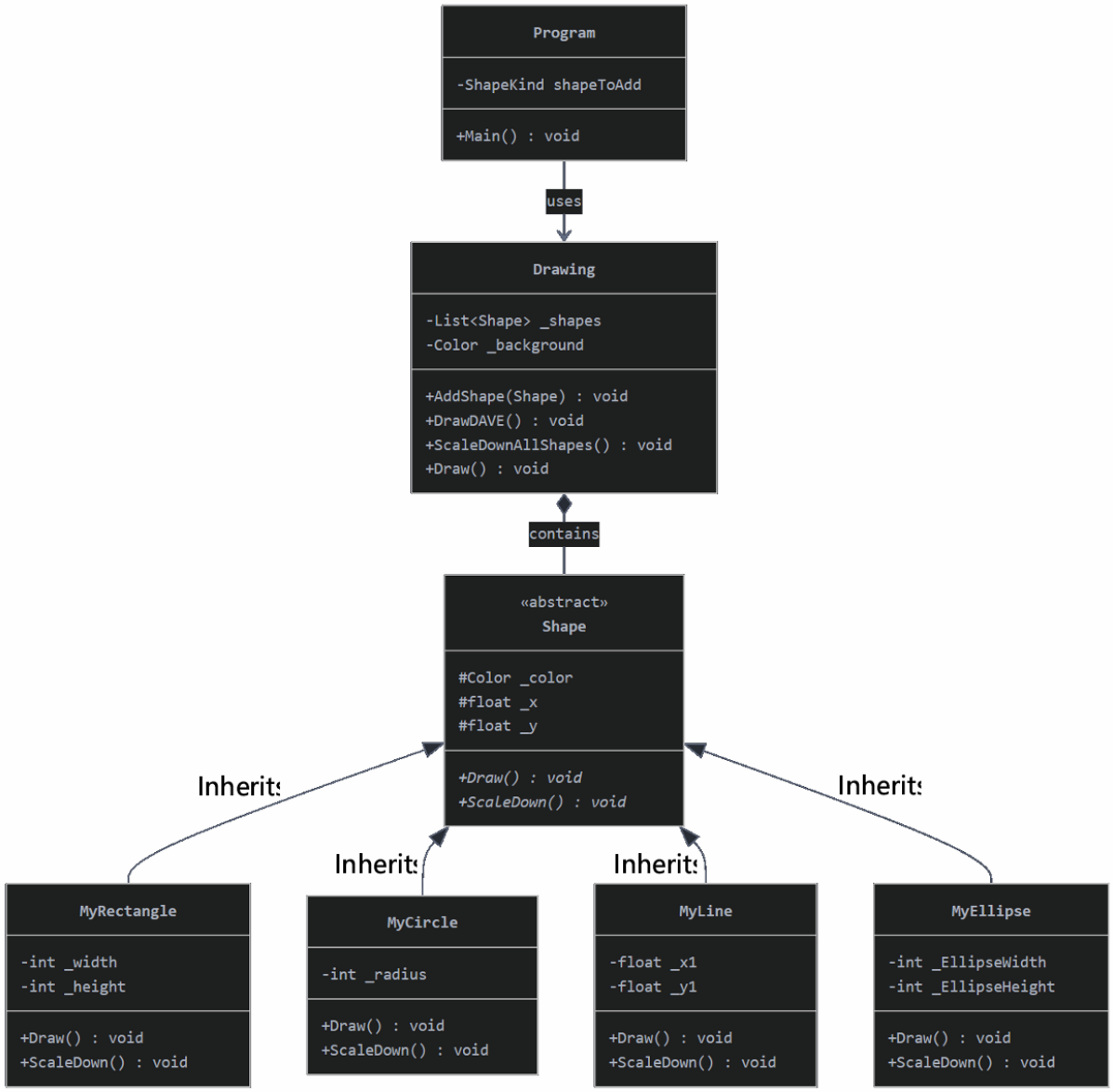

Fig.1: UML diagram.