Assignment 2
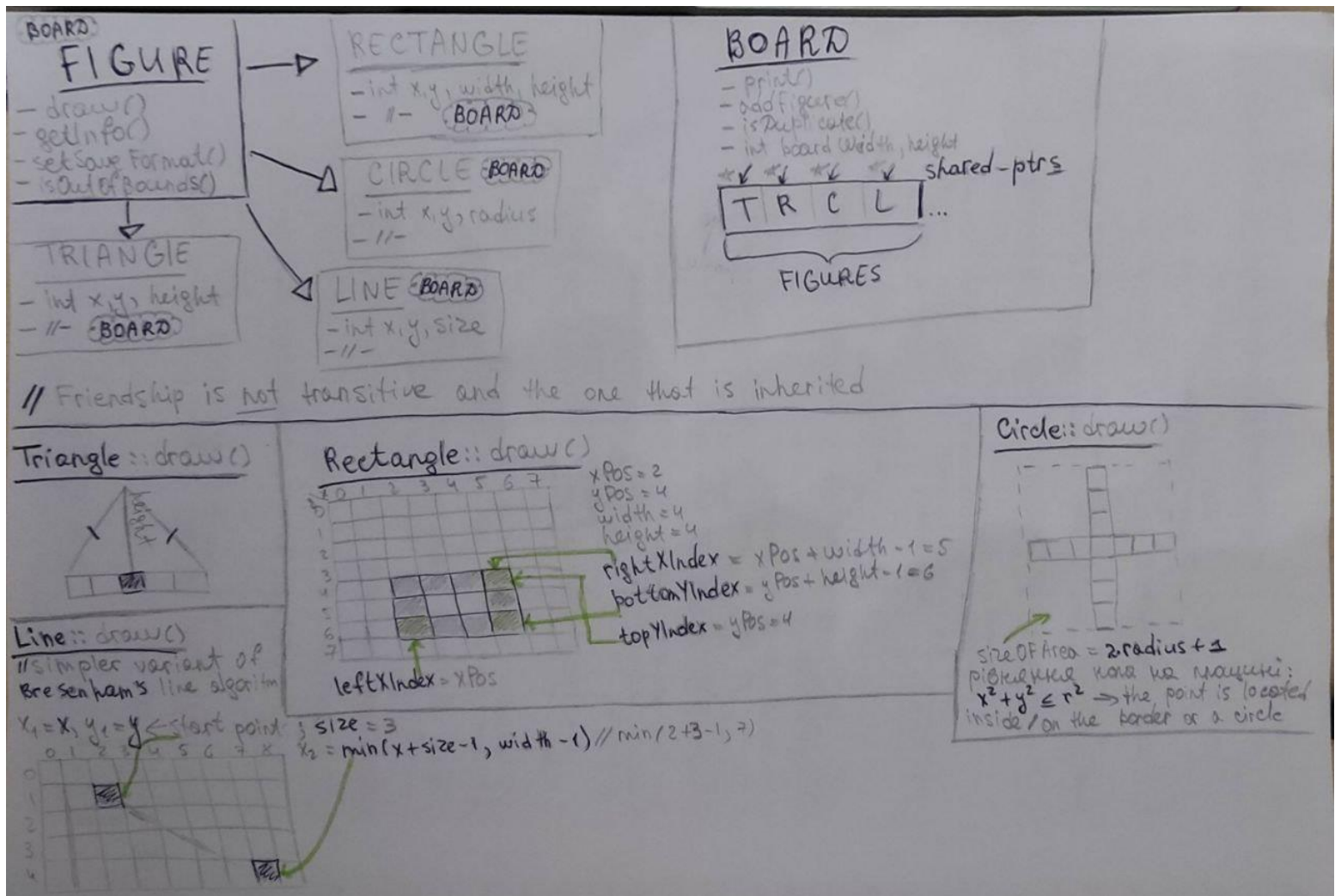
Student Name:

Olesia Mykhailyshyn

Group:

ОПД 2

Task:

This project is a console-based application for drawing various geometric shapes on a grid-based blackboard. The main goal is to implement a system that uses polymorphism and inheritance to support drawing multiple types of shapes while providing an interactive command-line interface (CLI) to manipulate these shapes. The application supports creating, listing, and managing shapes such as triangles, rectangles, circles, and lines.

This task leverages inheritance and polymorphism to create a flexible and extensible system where new shapes can be easily added by extending the Figure class. The CLI provides a simple interface for users to interact with the application.

System Model:

BOARD
FIGURE
- draw()
- getInfo()
- setSaveFormat()
- isOutOfBounds()

TRIANGLE
- int x, y, height
- // - BOARD

RECTANGLE
- int x, y, width, height
- // - BOARD

CIRCLE BOARD
- int x, y, radius
- // -

LINE BOARD
- int x, y, size
- // -

BOARD
- print()
- addFigure()
- isDuplicate()
- int board Width, height
- shared-ptrs

| T | R | C | L | ... |

FIGURES

// Friendship is not transitive and the one that is inherited

Triangle :: draw()

Rectangle :: draw()

xPos = 2
yPos = 4
width = 4
height = 4
rightXIndex = xPos + width - 1 = 5
bottomYIndex = yPos + height - 1 = 6
topYIndex = yPos = 4
leftXIndex = xPos

Line :: draw()
// simpler variant of Bresenham's line algorithm

$x_1 = x, y_1 = y$ start point
$x_2 = \min(x + size - 1, width - 1)$ // min(2+3-1, 7)

size = 3

Circle :: draw()

size OF Area = 2·radius + 1
$x^2 + y^2 \leq r^2 \Rightarrow$ the point is located inside / on the border of a circle

GitHub Link:

https://github.com/olesia-mykhailyshyn/Assignment_2_OOP.git

Solution Description:

The implemented software is a simple drawing application designed for drawing geometric shapes on a grid-based board using ASCII characters. It allows users to add, manipulate, and display shapes such as triangles, rectangles, circles, and lines through a command-line interface (CLI).

Figure Class Hierarchy: Defines a base class `Figure` and derived classes for each shape type (`Triangle`, `Rectangle`, `Circle`, and `Line`).

Board Class: Represents a grid of characters (`Board`) where shapes are drawn and displayed.

CLI Class: Provides a user interface for interacting with the board and shapes through text commands.

Command Handling and User Interaction:

The main application (`main.cpp`) initializes a `CLI` object and enters an infinite loop to continuously read and execute commands from the user.

Depending on the command, the corresponding method in `CLI` is called to perform the required operation.

For example, when a user enters the `add` command, the system prompts for additional parameters such as shape type, coordinates, and dimensions. Based on the shape type, a corresponding shape object is created and added to the board(main).

File Operations for Saving and Loading:

The system supports saving the current board state (along with all shapes) to a text file and loading it back. Each shape has a `getSaveFormat()` method that returns a string in a specific format to represent its state.

The `CLI::save` and `CLI::load` methods handle file I/O, ensuring that the board can be persisted between sessions(CLI).

Testing:

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
uuu
Unknown command.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
There are no any figures on the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
shapes
List of available shapes and their parameters:
1. Circle: x, y, radius
2. Rectangle: x, y, width, height
3. Triangle: x, y, height
4. Line: x, y, length

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
undo
There are no figures. Redo command can not be done.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
clear
There are no figures. Clear command cannot be done.
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
save
There are no figures. An empty file will be saved.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
load
The file is empty. No figures will be loaded.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
exit
Exiting the program.
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
add
Enter shapeName (triangle/rectangle/circle/line):
uuu
Enter x:
6
Enter y:
7
Enter height/radius/size:
5
Invalid shape name: uuu. Please enter a valid shape (triangle/rectangle/circle/line).

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
add
Enter shapeName (triangle/rectangle/circle/line):
triangle
Enter x:
8
Enter y:
6
Enter height/radius/size:
5
Successfully added triangle to the board.
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
add
Enter shapeName (triangle/rectangle/circle/line):
rectangle
Enter x:
12
Enter y:
13
Enter width:
4
Enter height:
6
Successfully added rectangle to the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
add
Enter shapeName (triangle/rectangle/circle/line):
circle
Enter x:
9
Enter y:
5
Enter height/radius/size:
3
```

```
Successfully added circle to the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
add
Enter shapeName (triangle/rectangle/circle/line):
line
Enter x:
1
Enter y:
9
Enter height/radius/size:
12
Successfully added line to the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
        *
      *   *


     *       *
       *
      * * *
     *   **
  ************
     *********



                ****
                *   *
                *   *
                *   *
                *   *
                ****
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
Figures on the board:
Triangle at (8, 6), height: 5
Rectangle at (12, 13), width: 4, height: 6
Circle at (9, 5), radius: 3
Line at (1, 9), length: 12

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
undo
The last added figure was deleted.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
        *
      *   *


     *      *
       *
      * * *
     *   **
    *       *
    *********




          ****
         *    *
         *    *
         *    *
         *    *
          ****
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
Figures on the board:
Triangle at (8, 6), height: 5
Rectangle at (12, 13), width: 4, height: 6
Circle at (9, 5), radius: 3

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
undo
The last added figure was deleted.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
        *
       * *
      *     *
     *         *
    *********



             ****
             *  *
             *  *
             *  *
             *  *
             ****
```

Figures saved to C:/KSE/OOP_design/Assignment_2/myFile.txt
save
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):

board.h board.cpp figure.h figure.cpp CLI.h CLI.cpp main.cpp ≡ myFile.txt ×
```
1    Triangle 8 6 5 0
2    Rectangle 12 13 4 6
3
```

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
clear
All shapes are removed from the blackboard. File is empty as well.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
There are no any figures on the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw

```
1  |
```

💡

Run   ⬛ Assignment_2 ✕

⟳ ⬛ ⬇ 🗑 ⋮

```
Figures saved to C:\KSE\OOP_design\Assignment_2\myFile.txt

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
clear
All shapes are removed from the blackboard. File is empty as well.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
There are no any figures on the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw
```

```
              *
          **     **
          *         *
          *         *

        *             *


          *         *
          *         *
          **     **
              *
Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
save
Figures saved to C:\KSE\OOP_design\Assignment_2\myFile.txt

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
exit
Exiting the program.
```

```
1        Circle 15 19 5 0
```

```
1    Circle -1 1 5 0
2    Circle 7 9 5 0
```

Run   Assignment_2 ×

C:\KSE\OOP_design\Assignment_2\cmake-build-debug\Assignment_2.exe

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
There are no any figures on the board.

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
load
Figure is out of bounds or larger than the board!
Error: Not all figures could be loaded from C:\KSE\OOP_design\Assignment_2\myFile.txt

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
list
Figures on the board:
Circle at (7, 9), radius: 5

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
draw

Enter command (draw/list/shapes/add/undo/clear/save/load/exit):
clear
All shapes are removed from the blackboard. File is empty as well.

Conclusions:

- **What was implemented**:

  - Developed a console-based drawing application to render various shapes (triangles, rectangles, circles, and lines) on a grid board.
  - Implemented OOP principles using inheritance and polymorphism.
  - Utilized `shared_ptr` for safe memory management of figures.

    - **Differences from the proposed model**:

  - Implemented some custom validations for shape parameters that were not defined in the original model.
  - Adjusted file loading and error-handling mechanisms based on real usage scenarios.

- **Additional Notes**:

  - The most challenging part was implementing the drawing algorithms for different shapes, as precise logic was required.
  - Relied on external resources like YouTube tutorials and algorithm websites to understand and implement shape rendering.

Appendices:

https://www.youtube.com/watch?time_continue=423&v=S_isDjezoz8&embeds_referring_euri=https%3A%2F%2Fwww.google.com%2Fsearch%3Fq%3Dhow%2Bto%2Bdraw%2Brectangle%2Bc%252B%252B%2Bby%2Bcoordinates%26oq%3Dhow%2Bto%2Bdraw%2Brectangle%2Bc%252B%252B%2Bby%2Bcoordinates%26gs_&source_ve_path=MTM5MTE3LDEzOTExNywyMzg1MQ --- drawing the rectangle

https://www.geeksforgeeks.org/program-print-circle-pattern/ --- circle

https://www.youtube.com/watch?v=HaZh4SVCXyg – circle

https://qna.habr.com/q/574775 -- line