The University of Melbourne
School of Computing and Information Systems
COMP90041 Programming and Software Development
Lecturers: Prof Udaya Parampalli & Dr Thuan Pham
Semester 1, 2021

Assignment 1
Due: 10am (AEST), April 19, 2021

1 Overview

In this assignment, you will write a simple console drawing program that implements some simplified features of a popular graphic application such as Microsoft Paint. Specifically, given some parameters (e.g., side length, alignment settings), the program draws isosceles right-angled triangles and squares on a drawing canvas. The program will also have a certain capability to implement some simple transformations like rotation and zooming and perform simple error checking. This assignment is designed in such a way that we can evaluate your knowledge on the following topics: 1) basic class design and implementation, 2) control flow structures (e.g., if-then-else, loops), and 3) basic I/O (e.g., Scanner class, formatted printing).

2 Your Task

Go to *https://classroom.github.com/a/RAVaxEzI* and accept the assignment. For details on how to check out the repository, make sure to consult the Lab 3 Materials[1]. Once you have cloned the repository, you will find four classes in it:

`ConsoleDrawing.java`
`DrawingCanvas.java`
`Triangle.java`
`Square.java`

`ConsoleDrawing.java` will be the main application containing your `main()` method. The other three classes will contain methods for setting up the drawing canvas, handling user inputs (e.g., information of the shapes to be drawn), and drawing the supported shapes. All these .java files are almost empty and you are asked to add constructors, instance variables, and methods to complete the program. See the application walk-through to understand the required features of the program.

For this assignment, **all user inputs will be assumed to be the correct data types;** such that if a String is expected then it is assumed a string will be entered, and the same for an integer. No error control for incorrectly entered data types are required... as of yet! .

---

[1]https://canvas.lms.unimelb.edu.au/courses/105405/assignments/200112

3 Application Walk-through

1. Your program must take three arguments which will be used to set up the initial drawing canvas. For example, the following command starts your program with a 10x6 drawing canvas and the background character is the minus sign.

    ```
    java ConsoleDrawing 10 6 -
    ```

    Note that this is a console drawing app so instead of using one pixel, we use one character as the measurement unit. A 10x6 drawing canvas is like a table with 6 rows and 10 columns in which each cell has enough space for only one printable character. Following is an example of a 10x6 canvas on which a 5x5 "square" has been drawn.

    ```
    AAAAA-----
    AAAAA-----
    AAAAA-----
    AAAAA-----
    AAAAA-----
    ----------
    ```

2. Your program will begin by displaying a welcome message. It will then display the initial drawing canvas settings as given in the program arguments.

    ```
    ----WELCOME TO MY CONSOLE DRAWING APP----
    Current drawing canvas settings:
    - Width: 10
    - Height: 6
    - Background character: -
    ```

3. The program will then display the main menu with a list of options for 1) drawing isosceles right-angled triangles, 2) drawing squares, 3) updating the drawing canvas, and 4) exiting the program.

    ```
    Please select an option. Type 4 to exit.
    1. Draw triangles
    2. Draw squares
    3. Update drawing canvas settings
    4. Exit
    ```

4. If an invalid option is provided, say the user types "5" and presses Enter, an error message will be displayed and the main menu is listed again.

    ```
    Unsupported option. Please try again!
    Please select an option. Type 4 to exit.
    1. Draw triangles
    2. Draw squares
    3. Update drawing canvas settings
    4. Exit
    ```

5. If the first option ("Draw triangles") is selected, by typing "1" and then pressing Enter, the program should call a corresponding method of the Triangle class to accept more user inputs and draw triangles accordingly. Specifically, it should first ask for the side length of an isosceles right-angled triangle, the printing character (which will be drawn inside the triangle on the canvas), and its alignment. Regarding error handling, you need to check if the canvas is big enough for the triangle. In case of alignment options, if an invalid one is selected, say "a" or "b", the default choice should be left alignment. Your program will then draw a triangle on the canvas if all settings are valid.

```
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
1
Side length:
8
Error! The side length is too long (Current canvas size is 10x6). Please try again.
Side length:
5
Printing character:
A
Alignment (left, middle, right):
left
AAAAA----
AAAA-----
AAA------
AA-------
A--------
---------
```

Please find below the program output if middle or right alignment has been selected. You can see that the triangles look the same but their positions on the canvas are different. If we use a tuple (**row_index, col_index**), where the indexes start from 1, to specify the top-left position of a shape on the drawing canvas, for all alignment settings, row_index equals 1. It means that all shapes should be drawn from the first row. The following formulas show how you should calculate the column index. Note that the same rule applies to squares.

```
Left alignment: col_index = 1
Middle alignment: col_index = (canvas_width - side_length) / 2 + 1
Right alignment: col_index = canvas_width - side_length + 1
```

```
(middle alignment)
--AAAAA---
--AAAA----
--AAA-----
--AA------
--A-------
----------

(right alignment)
-----AAAAA
-----AAAA-
-----AAA--
-----AA---
-----A----
----------
```

6. Your program will then ask if the user want to do some rotations by typing R (for a 90-degree clockwise rotation) or L (for a 90-degree anticlockwise rotation) and then pressing Enter. Note that the selection is case insensitive (i.e. R/r and L/l are accepted).

```
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
```

The following figures show the initial position of a left-aligned triangle and its new positions after being rotated 90 degrees, 180 degrees, 270 degrees, and 360 degrees, respectively. The same transformation rules apply to middle-aligned and right-aligned triangles. **Note that all transformations should not change the current alignment setting i.e. if a triangle or a square is left-aligned, the rotated triangle or the zoomed square must also be left-aligned**.

```
(A isosceles right triangle T at its initial position)
AAAAA-----
AAAA------
AAA-------
AA--------
A---------
----------

(90 degrees clockwise rotation of T)
AAAAA-----
-AAAA-----
--AAA-----
---AA-----
----A-----
----------

(180 degrees clockwise rotation of T)
----A-----
---AA-----
--AAA-----
-AAAA-----
AAAAA-----
----------

(270 degrees clockwise rotation of T)
A---------
AA--------
AAA-------
AAAA------
AAAAA-----
----------

(360 degrees clockwise rotation of T -- T is drawn at its initial position)
AAAAA-----
AAAA------
AAA-------
AA--------
A---------
----------
```

7. When users stop rotating the triangle, by typing any characters other than R and L and then pressing Enter, your program will ask if they want to draw another triangle.

```
Draw another triangle (Y/N)?
```

If Y is selected, the program should accept user inputs and draw another triangle. If N is selected, it should go back to the main menu. Otherwise, an error message should be displayed. Note that the selection is also case insensitive.

```
Draw another triangle (Y/N)?
A
Unsupported option. Please try again!
Draw another triangle (Y/N)?
N
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
```

8. In the main menu, if the second option (Draw squares) is selected, the program will follow a similar workflow when the first option (Draw triangles) is selected except that zooming (in and out) will be supported instead of rotations. **Note that each time a square is zoomed in/out, its side length will be increased/decreased by one character. Once the square reaches its limit i.e. its side length equals to min(canvas_width, canvas_height) or 1, users cannot make it bigger or smaller, respectively. If they try to do so, nothing changes.**

```
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
2
Side length:
5
Printing character:
B
Alignment (left, middle, right):
middle
--BBBBB---
--BBBBB---
--BBBBB---
--BBBBB---
--BBBBB---
----------
Type I/O to zoom in/out. Use other keys to continue.
I
--BBBBBB--
--BBBBBB--
--BBBBBB--
--BBBBBB--
--BBBBBB--
--BBBBBB--
Type I/O to zoom in/out. Use other keys to continue.
O
--BBBBB---
--BBBBB---
--BBBBB---
--BBBBB---
--BBBBB---
```

```
          ----------
          Type I/O to zoom in/out. Use other keys to continue.
          O
          ---BBBB---
          ---BBBB---
          ---BBBB---
          ---BBBB---
          ----------
          ----------
          Type I/O to zoom in/out. Use other keys to continue.
          Q
          Draw another square (Y/N)?
```

9. The third option in the main menu is for changing the current canvas settings. Once the settings have been changed, your program will print out a confirmation message and the newest settings.

```
          Please select an option. Type 4 to exit.
          1. Draw triangles
          2. Draw squares
          3. Update drawing canvas settings
          4. Exit
          3
          Canvas width: 20
          Canvas height: 20
          Background character: +
          Drawing canvas has been updated!

          Current drawing canvas settings:
          - Width: 20
          - Height: 20
          - Background character: +

          Please select an option. Type 4 to exit.
          1. Draw triangles
          2. Draw squares
          3. Update drawing canvas settings
          4. Exit
```

10. To exit the program, users can choose the fourth option from the main menu. And the program should display a goodbye message.

```
          Please select an option. Type 4 to exit.
          1. Draw triangles
          2. Draw squares
          3. Update drawing canvas settings
          4. Exit
          4
          Goodbye!
```

4 Example execution

Note that in the following example execution, we show both the input and output of the program. In the given test cases, the input and output are stored in different files. See Section 7 (Testing Before Submission) for more details.

```
----WELCOME TO MY CONSOLE DRAWING APP----
Current drawing canvas settings:
- Width: 10
- Height: 6
- Background character: -

Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
5
Unsupported option. Please try again!
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
1
Side length:
20
Error! The side length is too long (Current canvas size is 10x6). Please try again.
Side length:
5
Printing character:
A
Alignment (left, middle, right):
left
AAAAA-----
AAAA------
AAA-------
AA--------
A---------
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
R
AAAAA-----
-AAAA-----
--AAA-----
---AA-----
----A-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
R
----A-----
---AA-----
--AAA-----
-AAAA-----
AAAAA-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
R
A---------
AA--------
AAA-------
AAAA------
```

```
AAAAA-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
R
AAAAA-----
AAAA------
AAA-------
AA--------
A---------
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
A---------
AA--------
AAA-------
AAAA------
AAAAA-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
----A-----
---AA-----
--AAA-----
-AAAA-----
AAAAA-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
AAAAA-----
-AAAA-----
--AAA-----
---AA-----
----A-----
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
AAAAA-----
AAAA------
AAA-------
AA--------
A---------
----------
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
A
Draw another triangle (Y/N)?
Y
Side length:
6
Printing character:
B
Alignment (left, middle, right):
right
----BBBBBB
----BBBBB-
----BBBB--
----BBB---
```

```
----BB----
----B-----
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
----B-----
----BB----
----BBB---
----BBBB--
----BBBBB-
----BBBBBB
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
--------B
-------BB
-------BBB
------BBBB
-----BBBBB
----BBBBBB
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
L
----BBBBBB
-----BBBBB
------BBBB
-------BBB
--------BB
---------B
Type R/L to rotate clockwise/anti-clockwise. Use other keys to continue.
A
Draw another triangle (Y/N)?
N
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
2
Side length:
8
Error! The side length is too long (Current canvas size is 10x6). Please try again.
Side length:
4
Printing character:
C
Alignment (left, middle, right):
middle
---CCCC---
---CCCC---
---CCCC---
---CCCC---
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
I
--CCCCC---
--CCCCC---
--CCCCC---
```

```
--CCCCC---
--CCCCC---
----------
Type I/O to zoom in/out. Use other keys to continue.
I
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
Type I/O to zoom in/out. Use other keys to continue.
I
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
--CCCCCC--
Type I/O to zoom in/out. Use other keys to continue.
O
--CCCCC---
--CCCCC---
--CCCCC---
--CCCCC---
--CCCCC---
----------
Type I/O to zoom in/out. Use other keys to continue.
O
---CCCC---
---CCCC---
---CCCC---
---CCCC---
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
O
---CCC----
---CCC----
---CCC----
----------
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
O
----CC----
----CC----
----------
----------
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
O
----C-----
----------
----------
```

```
----------
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
O
----C-----
----------
----------
----------
----------
----------
Type I/O to zoom in/out. Use other keys to continue.
A
Draw another square (Y/N)?
N
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
3
Canvas width: 10
Canvas height: 10
Background character: *
Drawing canvas has been updated!

Current drawing canvas settings:
- Width: 10
- Height: 10
- Background character: *

Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
2
Side length:
6
Printing character:
E
Alignment (left, middle, right):
right
****EEEEEE
****EEEEEE
****EEEEEE
****EEEEEE
****EEEEEE
****EEEEEE
*********
*********
*********
*********
Type I/O to zoom in/out. Use other keys to continue.
A
Draw another square (Y/N)?
```

```
N
Please select an option. Type 4 to exit.
1. Draw triangles
2. Draw squares
3. Update drawing canvas settings
4. Exit
4
Goodbye!
```

5   Important Notes

Automatic tests will be conducted on your program by compiling, running, and comparing your outputs for several test cases with generated expected outputs. The automatic test will deem your output wrong if your output does not match the expected output, even if the difference is just having an **extra space or missing a colon**. Therefore, it is crucial that **your output follows <u>exactly</u> the same format shown in the examples above.**

Also, use **ONLY ONE** Scanner object throughout your program. Otherwise the automatic tests may cause your program to generate exceptions and terminate. The reason is that in the automatic test, multiple lines of test inputs are sent all together to the program. As the program receives the inputs, it will pass them all to the currently active Scanner object, leaving the rest of the Scanner objects nothing to read and hence cause a run-time exception. Therefore, it is important that **your program has only one Scanner object.** Arguments such as "It runs correctly when I do the manual test, but fails under the automatic test" will not be accepted.

6   Assessment

This project is worth 15% of the total marks for the subject.

Your Java program will be assessed based on correctness of the output as well as quality of code implementation.

7   Testing Before Submission

You will find all input files and the expected output files in the Projects page on Canvas for you to use on your own. **In this assignment, markers will use the same set of test data for marking**. You should use them to test your code carefully and then submit your code on GitHub. Note that each commit you make is recorded in your GitHub repository. Details of how the submission works can be found in Section 8.

To test your code by yourself:

1. Check your Java code files, and make sure you have the test input data files ready (e.g., "`input1.txt`").

2. Open a console, navigate to your project directory (where your .java classes reside), and run compile your program: `javac *.java` (this command will compile all your java file in the current folder).

3. Run command: `java ConsoleDrawing 10 6 - < input1.txt > my-output1.txt` (it runs the ConsoleDrawing using contents in "`input1.txt`" as input and write the output to `my-output1.txt`). Note that the exact arguments "10 6 -" must be passed to the program because the given output files (e.g., output1.txt, output2.txt) were captured with the same arguments.

4. Inspect the file `my-output1.txt` as it contains any errors your program execution may have encountered.

5. Compare your result with the provided output file `output1.txt`. Fix your code if they are different.

6. Repeat steps 3-5 for all given input and output pairs. When you are satisfied with your project, commit your changes to GitHub.

   Please follow the instructions in Lab-3 documentation if you want to use IDEs like Eclipse.

8 Submission

Your submission should have at least four Java source code files. You must name them `ConsoleDrawing.java`, `DrawingCanvas.java`, `Triangle.java` and `Square.java` (if you correctly cloned the assignment repository, these files should already be in your working directory). You can add more classes if you want but **please ensure that you do not include irrelevant files (e.g., unused classes or test files) in your submission**. You should verify your submission locally as described above before submitting your code to GitHub.

You can edit and re-submit your code as many times you want as long as you submit before the submission deadline of the assignment. **Any updates to your code on GitHub made after the submission deadline will incur a late-submission penalty. This means that even if you make a minor edit and re-submit a file after the deadline your entire submission will be subject to a late-submission penalty!**

The deadline for the project is **10am (AEST), April 19, 2021**. The allowed time is more than enough for completing the project.

What will be graded? The **last** version of your program committed to GitHub before the submission deadline. Submissions via email will not be accepted!

8.1 Late-Submission Penalties

There is a 20% penalty per day for late submissions.

For example, suppose your project gets a mark of 5 but changes are submitted within 1 day after the deadline, then you get **20% penalty** and the mark will be 4 after the penalty.
There will be **0 marks for your submission if you make changes after the 10am (AEST), April 23, 2021**.

9 Individual Work

Note well that this project is part of your final assessment, so copying, working together, sharing work (i.e. cheating) is not acceptable! Any form of material exchange, whether written, electronic or any other medium is considered cheating, as is copying from any online sources in case anyone shares it. Providing undue assistance is considered as serious as receiving it, and in the case of similarities that indicate exchange of more than basic ideas, **formal disciplinary action will be taken for all involved parties without exceptions! We have a sophisticated tool that undertakes deep structural analysis of Java code to identify regions of similarity**.