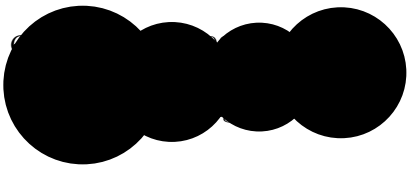
Assignment 6: Fancy Timer



Preparation

You are expected to be familiar with the following textbook sections and lectures *before* coming to the lab.

Textbook sections

- Chapter 6.1 - 6.4

Most relevant lectures

- L11

Introduction

You will write two programs: The first program will display any digit from 0-9 in a fancy way. The second program will be a countdown timer. You may work **alone or in pairs** for this assignment. In this assignment, you should write a javadoc comment at the top of each class and each method. You will need to write external documentation for the second program.

1 Fancy digits

In this part of the assignment, you will write a program called `Fancy.java` which displays a fancy version of the input digit.

1.1 Specifications

Input: The program should accept as input:

- a single digit (as an `int`) – the user should be prompted until the number entered is a single digit, however, you do not need to handle non-integer input
- a display character or sequence (as a `String`)

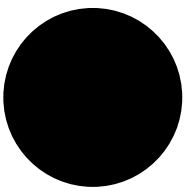
Output: A fancy version of the digit, composed of the display character or sequence.

Displaying fancy digits: If your display sequence is `"*"`, then here is what each of your fancy digits should look like:

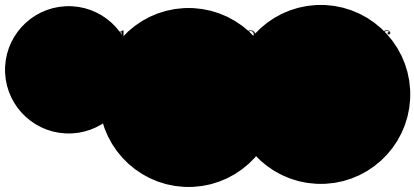
```
  * ***** ***** *   * ***** ***** ***** ***** ***** *****
  *     *     * *   * *     *           * *   * *   * *   *
  * ***** ***** ***** ***** *****       * ***** ***** *   *
  * *           *     *     * *   *     * *   *     * *   *
  * ***** *****       * ***** *****     * ***** ***** *****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

Note that you will only need to print **one digit at a time**, *not* all on the same line. They are displayed here on the same line just to save space.

If you look carefully at the digits, you will realize the following facts:



Assignment 6: Fancy Timer



1. Each fancy digit is made by printing five lines.
2. Each fancy digit is five characters wide.
3. There are only four possible types of lines that can make up a digit:
   (a) One * on the left, followed by four spaces.
   (b) Four spaces, followed by one * on the right.
   (c) One * on the left, followed by three spaces, followed by one * on the right.
   (d) A row of five *s.

Using methods: You must design and use the following methods to solve the problem. You will invoke `printDigit` from `main`. Within each method, you will need to invoke some of the other methods.

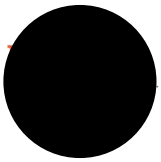
- A method to print a fancy version of any digit:
  - `public static void printDigit (int digitToPrint, String disp)` – displays a fancy version of `digitToPrint`, using the display sequence `disp`
- A display method for each of the ten digits:
  - `public static void printZero (String disp)` – displays a fancy version of the digit 0, using the display sequence `disp`
  - *methods for digits 1-8...*
  - `public static void printNine (String disp)` – displays a fancy version of the digit 9, using the display sequence `disp`
- Four methods to display each type of line a digit may be composed of:
  - `public static void printLeft(String disp)` – if `disp` is `"*"`, prints `"*    "`
  - `public static void printRight(String disp)` – if `disp` is `"*"`, prints `"    *"`
  - `public static void printLR(String disp)` – if `disp` is `"*"`, prints `"*   *"`
  - `public static void printRow(String disp)` – if `disp` is `"*"`, prints `"*****"`
- Method to generate spaces that match the length of `disp`:
  - `public static String getSpaces(int n, String disp)` – returns a `String` comprised of `n * disp.length()` spaces

1.2 Sample output

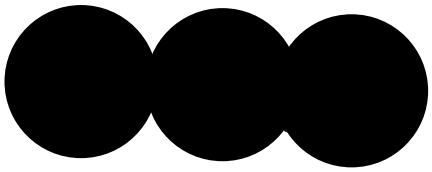
Sample 1

```
Please enter a digit to display:
6
Please enter a display character or String:
$$

$$$$$$$$$$
$$
$$$$$$$$$$
$$      $$
$$$$$$$$$$
```



Assignment 6: Fancy Timer



Sample 2

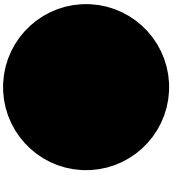
```
Please enter a digit to display:
4
Please enter a display character or String:
4

4   4
4   4
44444
    4
    4
```

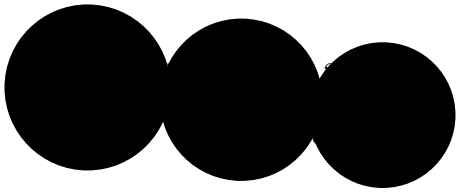
Sample 3

```
Please enter a digit to display:
2
Please enter a display character or String:
Hello

HelloHelloHelloHelloHello
                    Hello
HelloHelloHelloHelloHello
Hello
HelloHelloHelloHelloHello
```



Assignment 6: Fancy Timer



2 Countdown timer

You will write a program called `Timer.java`. You may write all your code in the main method. Do not rewrite the code needed to print fancy digits. You will need to use the methods you defined earlier in `Fancy.java`. As long as both java files are stored in the same directory, you may invoke the `printDigit` method inside your `Timer` program by typing `Fancy.printDigit`. The other methods can be invoked in the same way.

You will also need to write formal external documentation for this program. You may follow the example from class.

2.1 Specifications

Input: The program should accept a number of seconds from the user, and a display String.

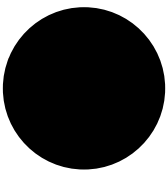
Output: The program should announce the beginning of the countdown. Then, **once per second**, it should display the remaining number of seconds, as a normal integer. However – if the remaining number of seconds is less than ten, display the fancy version of the digit instead!

2.2 Sample output

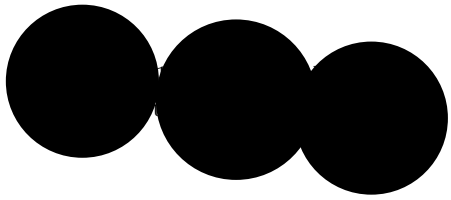
```
Please enter a number of seconds to count down:
14
Please enter a character or String to use in displaying the numbers:
%
Beginning countdown!
14
13
12
11
10
%%%%
%  %
%%%%
   %
%%%%

%%%%
%  %
%%%%
%  %
%%%%

%%%%
   %
   %
   %
   %
```



Assignment 6: Fancy Timer



//And all the fancy digits in between. And then...

```
    %
    %
    %
    %
    %

%%%%
%   %
%   %
%   %
%%%%
```

Submission

Recall that submission instructions are in the **Lab Guide**. Your group is required to submit **one** `.zip` folder (in one person's D2L dropbox) containing:

- the internally documented and properly styled source code files:
  - `Fancy.java`
  - `Timer.java`
- external documentation for the `Timer` program

If you are working in a group of two, the file should contain both of your names in the header. *Make sure both partners save copies of the finished code to their personal H: drive.*