Assignment #2
CSIS 1275 – Section 002
Instructor: Arezoo Ariafar

You are to begin the design and implementation of an Elevator class. You have been provided with a basic design and a minimal skeleton of the code. Here are some assumptions regarding the existing design and code:

1. The elevator cabin is a square that provides a capacity equal to its area. For example, an elevator cabin with side 2 provides the capacity of 4 passengers.
2. The floors of an elevator are numbered from 0 to MAX_FLOOR - 1.
3. An elevator is internally and visually represented using a two-dimensional array of characters holding special characters to show the structure itself, floor boundaries, as well as as the elevator cabin. Here is an example of a two-floor elevator with the capacity of 9 passengers, which is currently empty on the first floor:

#	#	#	#	#	#	#	#	#	#
#	E	E	E	|					#
#	E	E	E	|					#
#	E	E	E	|					#
#	#	#	#	#	#	#	#	#	#

   - 'E' character is used to illustrate empty rooms in the elevator cabin, and 'O' is used to show the occupied rooms.

4. The rooms inside the cabin are taken row by row from left to right, and are freed in reverse order.

```
                              Elevator
- MAX_Floor : int
- CAPACITY : int
- structure : char[][]
- currentFloor : int
- direction : boolean
- buttons : int[]
- floorButtons : boolean[][]
+ Elevator()
+ Elevator(max: int, cap: int)
+ goUp() : void
+ goDown() : void
+ getOn() : int
+ getOff() : int
+ printStructure() : void
+ stopAt(int floor) : void
- movePassengers(fromFloor: int, toFloor: int) : void
```

According to the above description, complete the implementation of Elevator.java to reflect the above design. Then, write an ElevatorDemo class to use the Elevator class following the instructions bellow:

1. Create two different elevators each instantiated by one of the Elevator constructors.
2. Get the arrangement of the floor buttons from user and set them appropriately.
3. Move the elevators, stop them on the requested floors, let the passengers get on and off the elevators.
4. Show the elevator moves on the screen.

Note:

To make a delay in your program output (wherever required), you call the sleep method of the Thread class, and send the delay time in milliseconds to it. For instance, the following instruction makes a 3 second delay in the program execution:

```
Thread.sleep(3000);
```