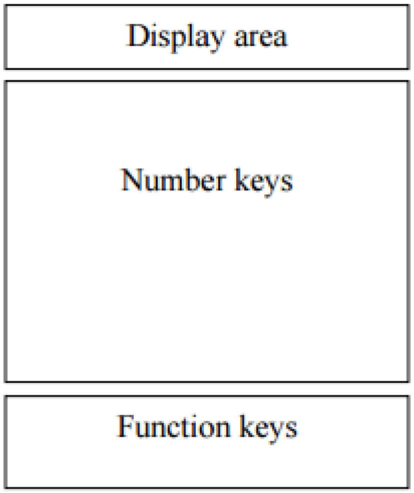
Consider a simple calculator interface. The interface looks as follows.



The “number keys” area contains the numeric keypad (0-9). The “function keys” contain four function keys: add, subtract, multiply and divide. To perform any function, the user has to “select” one number from the number keys, “select” the function key and finally “select” the other number (e.g. select 5, followed by “+” followed by 6 to perform 5+6). The result is displayed in the “display area”.

We want to design the calculator with a different input (selection) mechanism: instead of using mouse to select directly from the respective areas, we’ll design a “scanning input” mechanism; an alternative input technique useful for physically disabled users. In this, the selectable on-screen elements are periodically highlighted (i.e., change in color of the key). To select an element, the user has to wait for the element to be highlighted and then press “enter” key to select it. We’ll try to implement following variations of the above scheme.

1. The number keys will be periodically highlighted first. Once a number is selected (using the “enter” key), the function keys will be periodically highlighted. After a function key is selected, the periodic highlighting returns to the number keys for the selection of the second number. Thus, to perform the calculation 3+2, the user makes the following sequence of operation: wait for 3 to be highlighted -> press ‘enter’ -> wait for ‘+’ to be highlighted -> press ‘enter’ -> wait for 2 to be highlighted -> press ‘enter’.
2. Note that in the above input mechanism, the user cannot select multiple numbers before a function is selected. To avoid this, we can have both the number keys and the function Number keys Function keys Display area keys periodically highlighted simultaneously. To select a number, the user has to press “enter” key. To select a function, the user has to press the “space” key. Thus, to perform the calculation 33+25, the user makes the following sequence of operations: wait for 3 to be highlighted -> press ‘enter’ -> wait for 3 to be highlighted -> press

‘enter’ -> wait for ‘+’ to be highlighted -> press ‘space’ -> wait for 2 to be highlighted -> press ‘enter’ -> wait for 5 to be highlighted -> press ‘enter’.

Implement the calculator with the above input methods using the JAVA concurrency constructs and the SWING library for the interface. Note the following.

i) Highlighting can be implemented by changing color of the key

ii) A key should be highlighted for a specific time interval (you can fix it. However, it should be long enough to allow the user time to select the key).

iii) The highlighting is periodic (i.e., highlighter moves from one key to the next till the program terminates).

iv) You can have explicit “stop” mechanism (a button) to indicate the end of selection before the result is displayed. Note that such buttons will also be selected through scanning input only