Aima khan
2023F-BSE-064

LAB # 4

Start, Sleep and Stop methods of multithreading

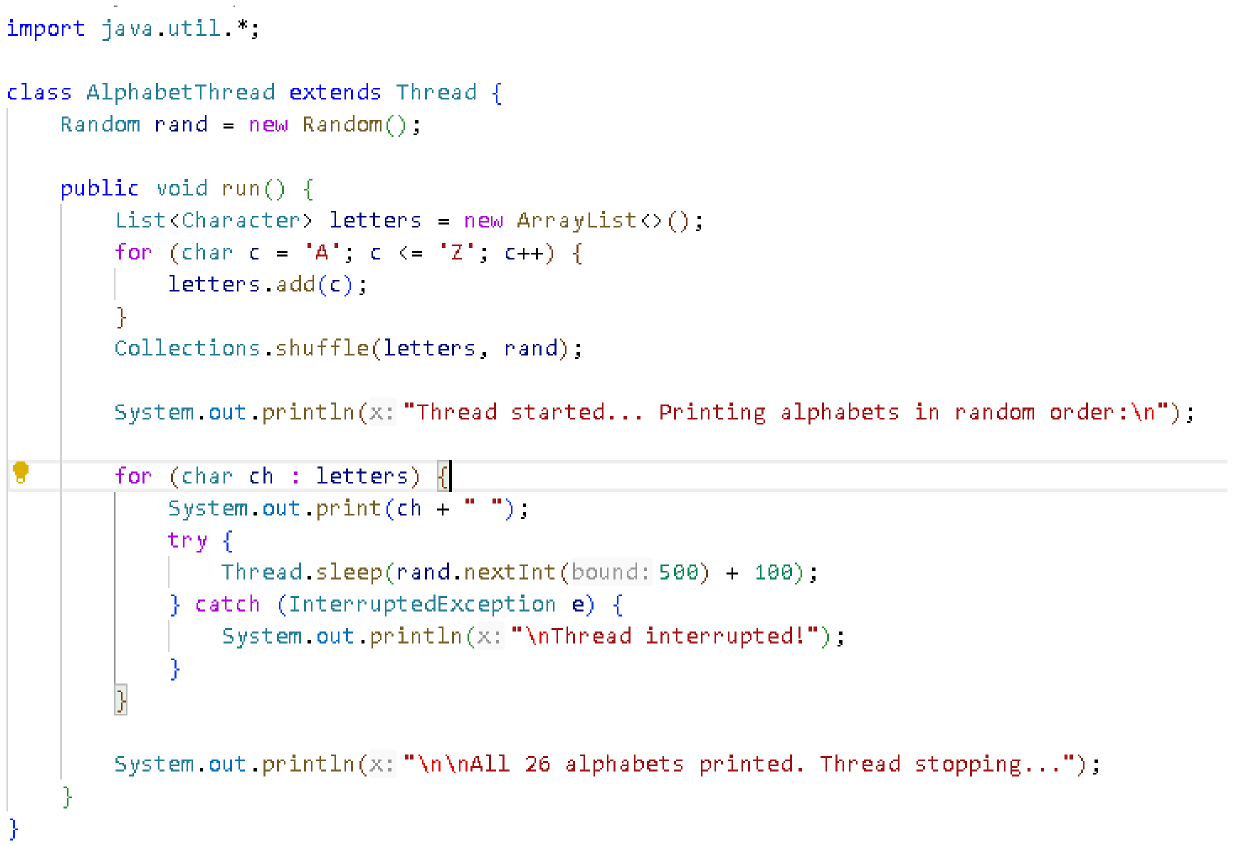
OBJECTIVE:

Understanding concurrency by implementing start, sleep and stop thread functions.

Lab Task:

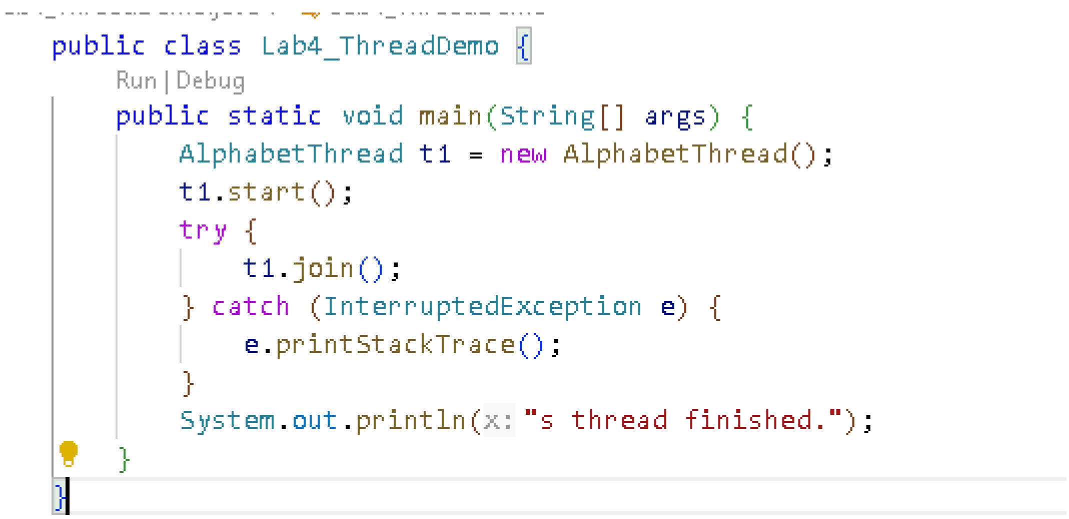
By using start, stop and sleep methods of threading, print alphabets of English from A-Z. (Hint: use math.random method for getting random numbers and then convert them into characters, print 26 characters under rum method loop with fluctuating visualization through sleep method).

CODE:



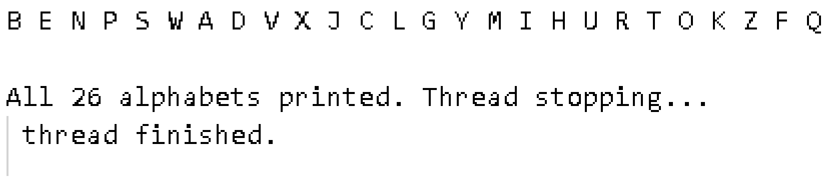
```
import java.util.*;

class AlphabetThread extends Thread {
    Random rand = new Random();

    public void run() {
        List<Character> letters = new ArrayList<>();
        for (char c = 'A'; c <= 'Z'; c++) {
            letters.add(c);
        }
        Collections.shuffle(letters, rand);

        System.out.println(x: "Thread started... Printing alphabets in random order:\n");

        for (char ch : letters) {
            System.out.print(ch + " ");
            try {
                Thread.sleep(rand.nextInt(bound: 500) + 100);
            } catch (InterruptedException e) {
                System.out.println(x: "\nThread interrupted!");
            }
        }

        System.out.println(x: "\n\nAll 26 alphabets printed. Thread stopping...");
    }
}
```

Aima khan
2023F-BSE-064



```
public class Lab4_ThreadDemo {
    Run | Debug
    public static void main(String[] args) {
        AlphabetThread t1 = new AlphabetThread();
        t1.start();
        try {
            t1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println(x: "s thread finished.");
    }
}
```

OUTPUT:



```
B E N P S W A D V X J C L G Y M I H U R T O K Z F Q

All 26 alphabets printed. Thread stopping...
 thread finished.
```