TP01

Multithreading

Remark

- Thread: sub process of execution of a program.
- Use library JnativeHook at https://github.com/kwhat/jnativehook

Threads

Consider when you want 2 methods execute at the same time. It is called asynchronous execution. In Java, we can do it using Threads. There are 2 ways for implementing threads in Java.

1. Inheriting Thread class:

```
public class SubclassThread extends Thread {
    @Override
    public void run() {
        System.out.println("Running in thread...");
    }

    public static void main(String[] args) {
        SubclassThread s = new SubclassThread();
        s.setDaemon(true);//run in background
        s.start();//start thread
        for(int i=0;i<10;i++)
        System.out.println(i+". Doing other code in main thread...");
    }
}
```

2. Implement Runnable interface:

```
public class ImplRunnableInt implements Runnable{
    @Override
    public void run() {
        System.out.println("Running in thread...");
    }
    public static void main(String[] args) {
        ImplRunnableInt ir = new ImplRunnableInt();
        Thread th = new Thread(ir);
        th.setDaemon(true);
        th.start();
        for(int i=0;i<10;i++)
        System.out.println(i+". Doing other code in main thread...");
    }
}
```

TP01.1. Stop Hits

Create a program to repeat display the word "HitMe!" until user presses ENTER.

Example:

HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!
HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!
HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!

Thanks you!

TP01.2. Stop Hits 2

Create a program to repeat display the word "HitMe!" until user presses ENTER.

If user type any other character other than ENTER, the program will display the character that user input instead of "HitMe!"

Example:

HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!
HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!
HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!HitMe!
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbbbbiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiieeeee
eeeeeeeeeeeeeeeeeeeeoooooooooooooooooosssssssssssssssssssssssssss…………………….eeeeeeeeeeeeeeeeeeeeeeei
iiiiiiiiiiiiiiiiiixxxxxxxxxxxxxxxxxxxxxxooooooooooooooooooooooooiiiiiiiiiiiieeeeeeeeeeeeeer444444444444wwwwwww
wwwww49333333333333333333333333344444444444444444444

Thanks you!

TP01.3. Thread generator

Create a program to read input from user the number of threads to be created. If user type number 4, it will create 4 threads.
Each thread, displays its index (ex: 0 1 2 3) number.

When executing the program, take note the result it displayed. Is it display orderly such as 0 1 2 3 … ?

If not, why? How could we fix it? Implement your correction.

TP01.4. Thread generator

Create a program to read input from user the range (start, end) and then create thread(s) to display and count prime numbers in this range. Each thread responsible maximum 100 numbers in the range. Each thread should have a name as example "t0-" means thread at index 0.
Example:

```
Input start: 80
Input end: 350
Running 3 threads.
t1-151 t0-251 t0-257 t2-83 t0-263 t0-269 t1-157 t0-271 t2-89 t0-277 t1-163 t0-281 t2-97
t0-283 t1-167 t0-293 t2-101 t2-103 t2-107 t0-307 t1-173 t0-311 t2-109 t0-313 t0-317 t1-
```

```
179 t0-331 t0-337 t2-113 t0-347 t1-181 t0-349 t2-127 t1-191 t2-131 t1-193 t2-137 t2-139
t1-197 t2-149 t1-199 t1-211 t1-223 t1-227 t1-229 t1-233 t1-239 t1-241

Sum of primes = 48
```

TP01.5. Brain Test

Create a program to display 2 numbers A and B where A, B are positive and less than 10. Then, ask user to input the result of addition between A and B. User has 2 seconds to input the answer. If the time passed, user still not yet answer, the program will stop and display the test result (count number of correct answer and wrong answer).

If user input the result, we validate the result (correct/incorrect) and then continue generating and asking the same question until user miss input or reach 10 times.

Ex:

1 + 3 = ?
4

5 + 9 = ?
12

Result: correct= 1 ➔ baby brain

4 + 3 = ?
7

4 + 6 = ?
10

9 + 9 = ?
18
…

Result: correct= 10 ➔ master brain