Entities

=====

Value:

Value is an entity with two fields

1. Integer

2. boolean

Classes:

Randomizer

Prime

Randimizer is act as a Producer where the Prime is the receiver with the help of distributed queue.



Considering the above diagram Randomizer is Thread1& Prime is the Thread2.

Randomizerhas a method randomGenerate which is responsible for generate the random integers& put into the distributedQueue.

public Queue<Integer> randomGenerate() {

Queue<Integer> queue = new PriorityQueue<>();

int num;

Random rand = new Random();

for (int k = 0; k < 10; k++) {

num = rand.nextInt(Integer.MAX\_VALUE);

queue.add(num);

}

return queue;

}

Prime has a method primeCheck which is responsible for receiving the distributedQueue with Integers& returning the distributedQueue with list if entities “PrimaryQueue<Value>”.

public Queue<Value> primeCheck(Queue<Integer> q) {

boolean flag;

Iterator<Integer> itr = q.iterator();

while (itr.hasNext()) {

Value val;

int item = (int) itr.next();

int srt = (int) Math.sqrt((double) item);

if (item == 1) {

flag = true;

} else {

flag = true;

for (int i = 2; i <= srt; i++) {

if (item % i == 0) {

flag = false;

}

}

}

val = new Value(item, flag);

primeQueue.add(val);

}

return primeQueue;

}

Passing the Distributed Queue from function randomGenerate() to Prime; common data structure between two threads, a communication channel established.

Public static void main(String[] args) {

Randomizer random = new Randomizer();

Queue<Value> outputQueue = new Prime().primeCheck(random.randomGenerate());

System.out.println(outputQueue);

}

Sample Output:

[Value{item=275300513, isPrime=false}, Value{item=442505059, isPrime=false}, Value{item=340076221, isPrime=false}, Value{item=1225145492, isPrime=false}, Value{item=1507874643, isPrime=false}, Value{item=821204543, isPrime=false}, Value{item=846551000, isPrime=false}, Value{item=1674768545, isPrime=false}, Value{item=1310603049, isPrime=false}, Value{item=1570561762, isPrime=false}]