CODE OUTPUT

From Little’s Law:

Delay = Average Queue Length / probability of arrival:

|  |  |  |  |
| --- | --- | --- | --- |
| Probability Of Arrival | Probability of Departure | Average Queue Length | Delay |
| 0.1 | 0.6 | 0.9262 | 9.262 |
| 0.2 | 0.6 | 1.2156 | 6.0780 |
| 0.3 | 0.6 | 1.5504 | 5.1680 |
| 0.4 | 0.6 | 2.0747 | 5.1866 |
| 0.5 | 0.6 | 4.0340 | 8.0680 |
| 0.6 | 0.6 | 69.0872 | 115.1453 |

**public class** SingleServerQueue {  
  
 **static** LinkedList<Packet> *singleServerQueue*;  
 **static** LinkedList<Packet> *removedPackets*;  
  
 **public static void** main(String[] args) {  
  
 Random arrivalGenerator = **new** Random();  
  
 *singleServerQueue* = **new** LinkedList<Packet>();  
 *removedPackets* = **new** LinkedList<>();  
 **long** endTime = System.*currentTimeMillis*() + 100;  
  
 **for** (**int** i = 10; i > 0; i--) {  
  
 System.***out***.println(**"PROBABILITY OF ARRIVAL = "**+(10-(i-1))+**"/10"**);  
 ArrayList<Integer> queueLengths = **new** ArrayList();  
 **while** (System.*currentTimeMillis*() < endTime) {  
 queueLengths.add(*runSimulation*(i, arrivalGenerator, endTime));  
 }  
 **double** sum = 0;  
 **for** (**int** x :queueLengths) {  
 sum+=x;  
 }  
 System.***out***.println(**"AVERAGE QUEUE LENGTH: "**+ sum/queueLengths.size());  
 System.***out***.println();  
 endTime = System.*currentTimeMillis*() + 10;  
 }  
 }  
  
 **private static int** runSimulation(**int** i, Random arrivalGenerator, **long** endTime) {  
 *processAndDepart*(arrivalGenerator);  
 **if** (arrivalGenerator.nextInt(i) == 0) {  
 Packet arrived = **new** Packet();  
 *singleServerQueue*.add(arrived);  
 *processAndDepart*(arrivalGenerator);  
 }  
 **return** *singleServerQueue*.size();  
 }  
  
 **private static void** processAndDepart(Random arrivalGenerator) {  
 **if** (!*singleServerQueue*.isEmpty()) {  
 **if** (arrivalGenerator.nextInt(6) == 0) {  
 Packet removed = *singleServerQueue*.remove();  
 *removedPackets*.add(removed);  
 }  
  
 }  
 }  
   
}

**public class** Packet{  
  
 **long entryTime**;  
 **long exitTime**;  
  
 **public** Packet(){  
 **entryTime** = System.*currentTimeMillis*();  
 }  
 **public void** exitPacket(){  
 **exitTime** = System.*currentTimeMillis*();  
 }  
 **public long** getDelay(){  
 **return exitTime** = **entryTime**;  
 }  
}