Class 28 Notes
Queues for Simulation

The next example uses a queue to model and simulate a customer waiting line at an ATM machine.

During lunch hour, the ATM machine in a large office complex is in heavy demand. Customers complain that the waiting time is much too long. The local bank is considering the addition of a second machine. But first, the bank needs a few statistics to justify the cost.

Problem
Simulate a waiting line (a queue) at the ATM machine, minute by minute, for a period of one hour. Make the following assumptions:

- With **equal probability**, a customer spends:
  one minute,
  two minutes, or
  three minutes
  at the ATM machine.

.
- During any minute:

no customers arrive	(50% chance),
one customer arrives	(40% chance), or
two customers arrive	(10% chance).

As customers arrive, they are placed in a queue

At the end of an hour, display the following summary statistics:

- the number of customers served, that is, the number who accessed the ATM machine,
- the average time a that customer waits in queue before being served, and
- the number of customers that remain in the queue at the end of the simulation.

Assume that the ATM is available when the simulation begins and that no customers are waiting.

OK – Let's see how this will work. We assume that:
During any minute:

• no customers arrive	(50% chance),
• one customer arrives	(40% chance), or
• two customers arrive	(10% chance).

How do we determine the number of customers that arrive??????

To determine the number of arrivals in any minute, **generate a random number 0…9.**

- The probability that this random integer is 0, 1, 2, 3, or 4 is 0.50. (50%) ---No arrivals
- The probability that the random number is 5, 6 ,7, or 8 is 0.40. (40%) --- One arrival
- The probability that the number is 9 is 0.10. – 2 arrivals(10 %)

So, at any particular minute,

- if I generate 0,1,2,3 or 4→ No arrivals, and there's a 50% chance that happens
- if I generate 5,6,7 or 8 → one arrival, and there's a 40% chance that happens
- if I generate 9 →two arrivals, and there is just 10% chance of generating a 9

With equal probability, a customer spends:

- one minute,
- two minutes, or
- three minutes

at the ATM machine.

.

Again, generate a random number: 1 ,2 or 3 as the time spent at the ATM

So,at any particular minute, we will have to determine

- How many customers arrive…0, 1, or 2 customers
- How long each customer takes at the ATM …1,2 or 3 minutes

And, these are just random numbers as described above.

- A random number for the number of arrivals and
- a random number for the time a customer hogs up the ATM

Before looking at the algorithm and then the code I will do the simulation by hand.

We will need to keep track of

- The current time, that is the minute 0..59
- The number of customers served
- The "next time" the ATM is available. For example, if the current time is minute 2 and the current customer takes 3 minutes at the ATM then the next available time is 2+ 3 = 5

I will not do it by hand for 60 minutes. To see how this works, a 6 minute simulation is plenty.
So here goes. At the end of 6 minutes, I will calculate:

- The number served
- The average waiting time
- The number of customers remaining in the queue

Time = 0

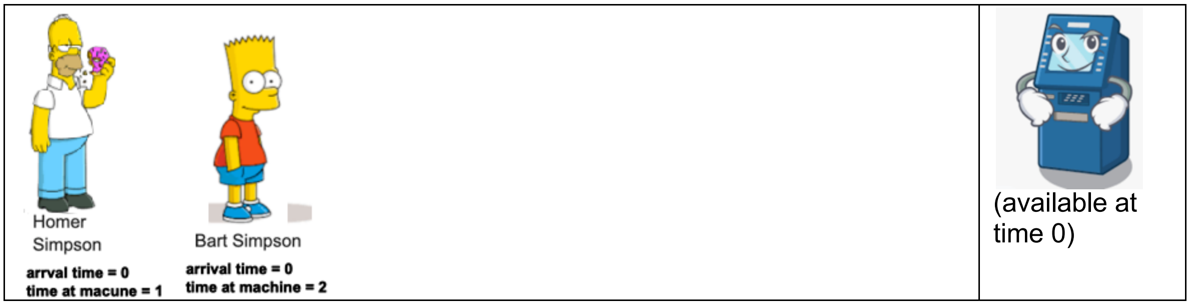
time=0	**ATM available at time = 0**
	Customers served = 0
	Total wait time = 0

Two arrivals: Bart and then Homer. (Two arrivals were obtained using a random number)

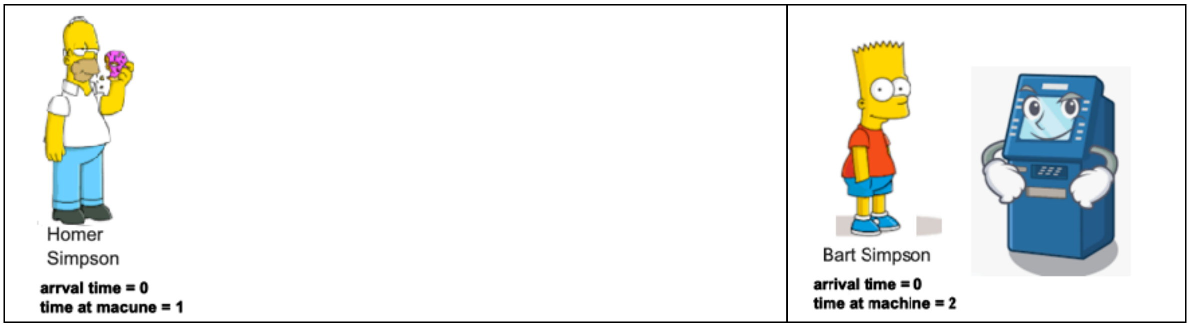
Notice **each customer is marked/labeled with**

- the arrival time (It is time = 0)
- the time at the ATM (Bart takes 2 minutes, Homer takes 1: using a random number)

Place each in the queue



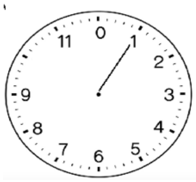
The ATM is available so Bart goes to the ATM, leaving Homer in the queue. (Dad gets no respect!)



Update the statistics

- ATM will be available again at time 2, since it is time = 0 and Bart takes 2 minutes
- Increment the number of customers served (It is now 1, just Bart)
- Bart did not wait, so total wait time is still 0

Time = 1

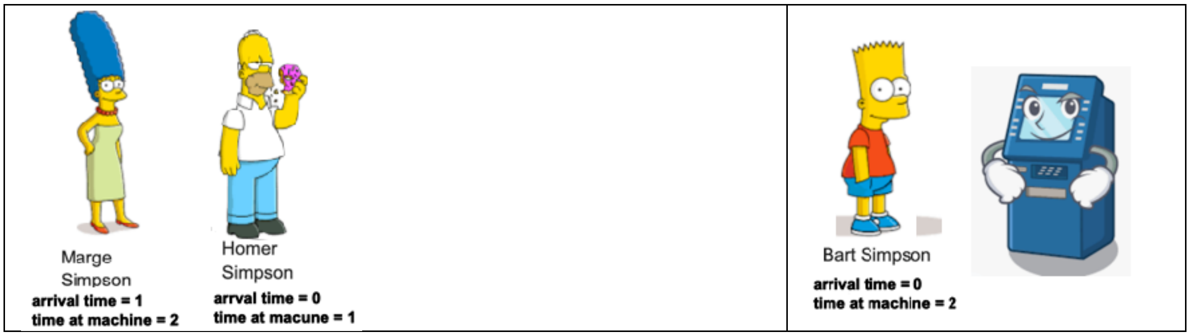
 time=1	ATM available at time = **2**
	Customers served = **1**
	Total wait time = **0**

One arrival: Marge→ (obtained using a random number)

Notice each customer – Homer and Marge-- is marked with

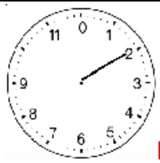
- the arrival time (Marge's arrival time is time = 1; Homer's was 0)
- the time at the ATM (Marge takes 2 minutes : using a random number)

Place Marge in the queue.



The ATM is not available until time = 2. (Bart is still there).
So no movement from the queue.

Time = 2

 time=2	ATM available at time = 2
	Customers served = 1
	Total wait time = 0

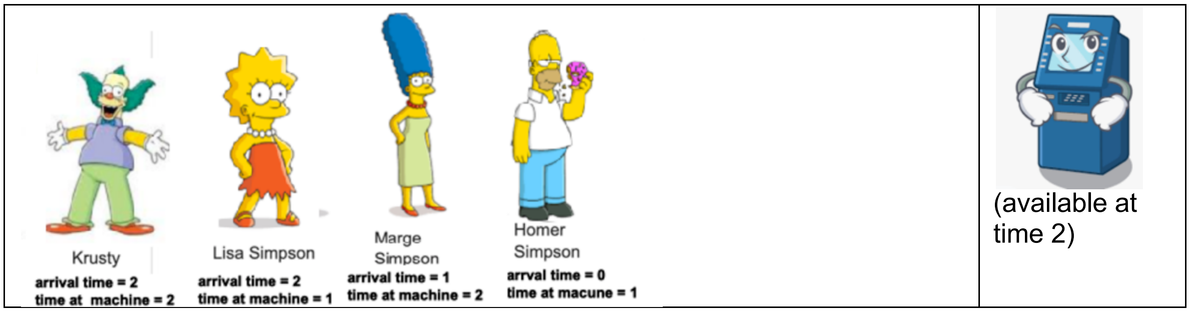
Two arrivals: Lisa, and Krusty,

Notice each customer – is marked with

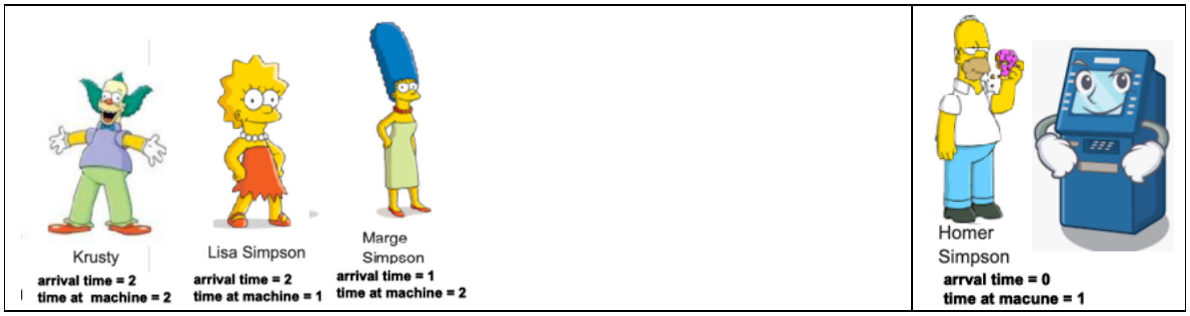
- the arrival time
- the time at the ATM

  Lisa takes 1 minutes, Krusty 2 : using a random number

Place the two new customers into the queue in the queue.
Notice **The ATM is available at time 2.**



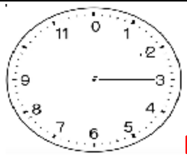
The ATM is available so Homer goes to the ATM:



Update the statistics:

- ATM will be available again at time 3, since Homer takes 1 minute
- Increment the number of customers served (It is now 2 → Bart and Homer)
- Homer entered at time 0, it is now time 2, so Homer had to wait 2 minutes.
  Add 2 to the total wait time, making it 2

Time = 3

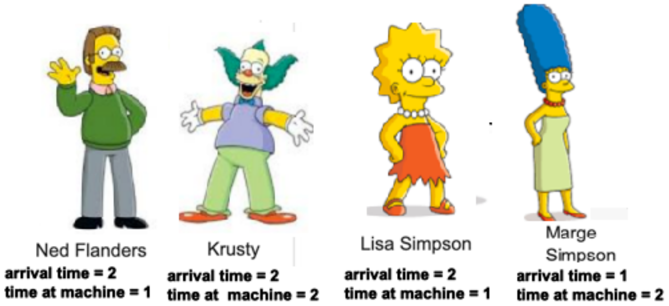
 time=3	ATM available at time = 3
	Customers served = 2
	Total wait time = 2

One arrival: Ned Flanders (obtained using a random number)

Notice Ned is labeled

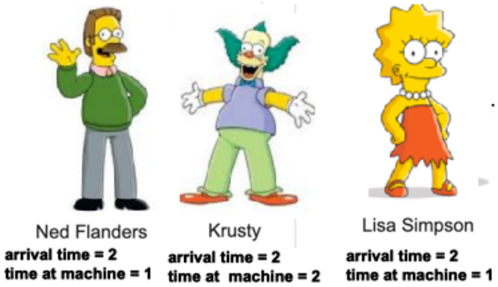
- with arrival time (time = 2)
- and time at the ATM (1 minute)

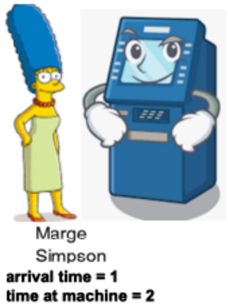
Place Ned Flanders into the queue.



Available at time = 3

The ATM is available. So Marge goes to the ATM. Leaving three in the queue





Available at time = 3

Update the statistics:

- ATM will be available again at time 5, since Marge takes 2 minutes
- Increment the number of customers served (It is now 3) – Bart, Homer, and Marge
- Marge arrived at time 1, it is now time 3 so Marge had to wait 2 minutes
  add 2 to the total wait time, making it 4

Time = 4

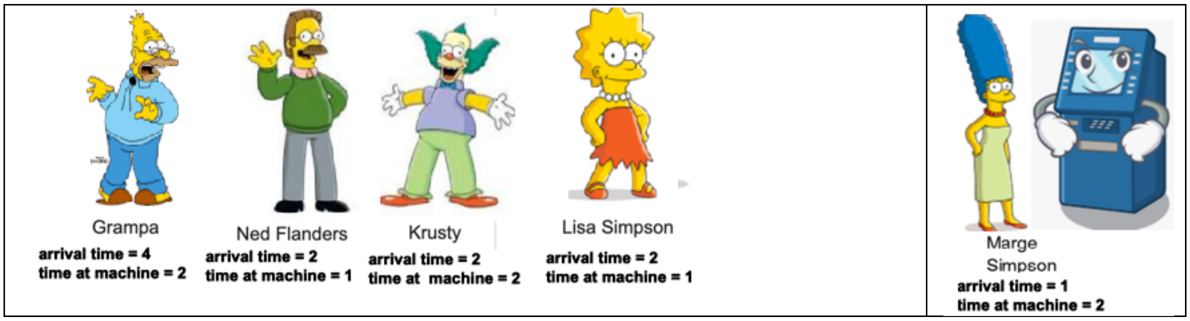
time=4	ATM available at time = **5**
	Customers served = 3
	Total wait time = 4

One arrival: Grampa Simpson

Notice each customer – is marked with

- the arrival time
- the time at the ATM (Grampa Simpson takes 2 minutes – random number)

Place the new customer (Grampa Simpson) into the queue.



The ATM is not available until time = 5. Marge is still at the ATM.
So, no one comes off the queue.

Time = 5

time=5	ATM available at time = **5**
	Customers served = 3
	Total wait time = 4

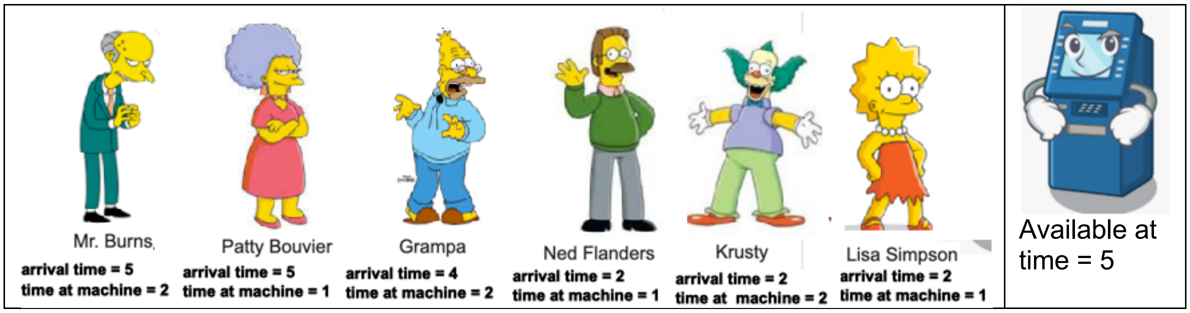
Two arrivals: Patty Bouvier and Mr. Burns

Notice each customer – is marked with

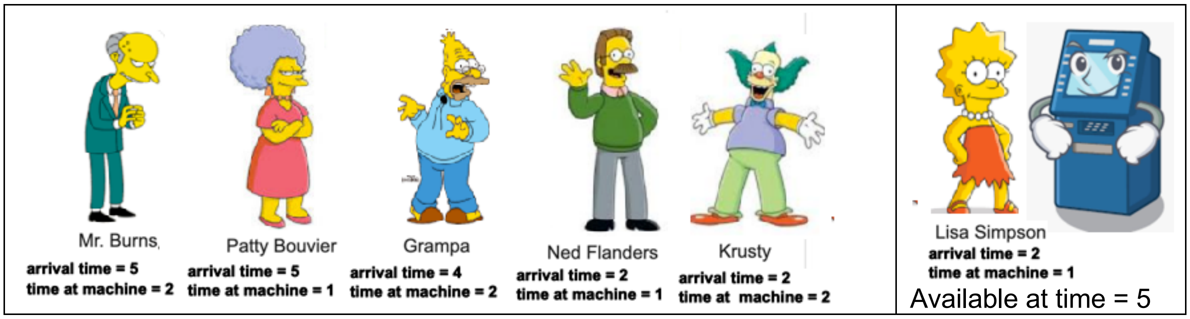
- the arrival time
- the time at the ATM

  Patty takes 1 minutes, Mr. Burns takes 2 : using a random number

Place the two new customers into the queue.
Notice The ATM is available at time 5. Marge is done.



Since the **ATM is Available**, Lisa goes to the ATM. So, five people are left in the queue.



Update the statistics:

- ATM will be available again at time 6, since Lisa takes 1 minute
- Increment the number of customers served → It is now 4 – Bart, Homer, Marge, Lisa
- Lisa arrived at time = 2, it is now time = 5, so Lisa had to wait 3 minutes
  add 3 to the total wait time, making the total wait time 7

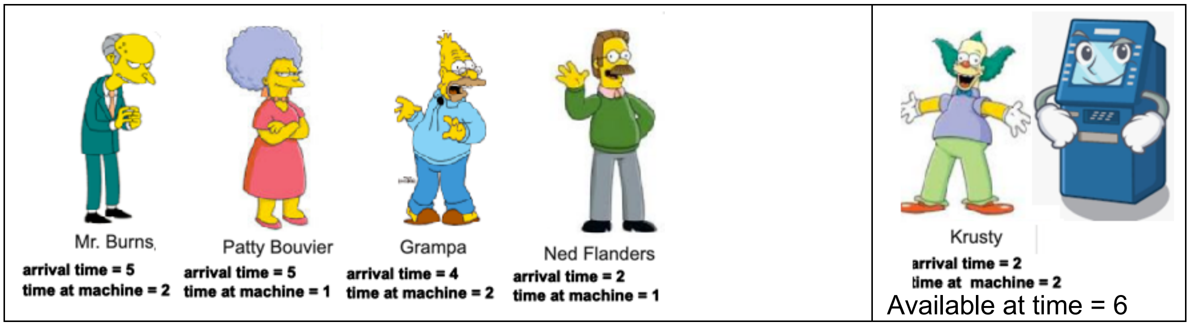
Time = 6

time=6	ATM available at time = 6
	Customers served = 4
	Total wait time = 7

No Arrivals at time 6 (obtained with a random number).
So no one enters the queue

The ATM is available

Krusty moves to the ATM, leaving four people in the queue



Update the statistics:

- ATM will be available again at time 8, →since Krusty takes 2 minutes
- Increment the number of customers served → It is now 5
- Krusty entered at time = 2, it is now time = 6 so Krusty had to wait 4 minutes add 4to the total wait time, making it 11

Time is 6 so simulation is DONE

Summary statistics:

- **Number of customers served : 5**
- **Average wait: Total waiting time/ number served →11/5 = 2.2 minutes**
- **Customers remaining in the queue: 4**

That's how the simulation works and that is how the program will work.

Before looking at an algorithm that simulates the comings and goings of customers at an ATM machine, I will **design a class that models an ATM customer.**

A customer is marked with his/her arrival time and how much time he/she spends making an ATM transaction (service time). So those are the fields for a Customer object. Methods are constructors, getters, and setters. The one argument constructor determines the service time (1,2, or 3 minutes).

```
public class Customer
{
    private int arrivalTime;       // 0..59, the minute when a customer arrives
    private int serviceTime;       // 1, 2, or 3 minutes

    public Customer()              // default constructor
    {
        arrivalTime = 0;
        serviceTime = 0;
    }

    public Customer(int arrTime)            // one argument constructor
    {
         arrivalTime = arrTime;
         Random rand = new Random();
         serviceTime = rand.nextInt(3)+1;  // 1, 2, or 3 minutes
     }

     public void setArrivalTime(int arrTime)
     {
         arrivalTime = arrTime;
     }

    public int getArrivalTime()
    {
        return arrivalTime;
    }

    public void setServiceTime(int ser)
    {
        serviceTime = ser;
    }

    public int getServiceTime()
    {
        return serviceTime;
    }
}
```

The algorithm that simulates an ATM waiting line uses a loop that ticks through a sixty minute simulation. This is exactly what I did above in pictures. If you understood the picture simulation, you will understand the algorithm.

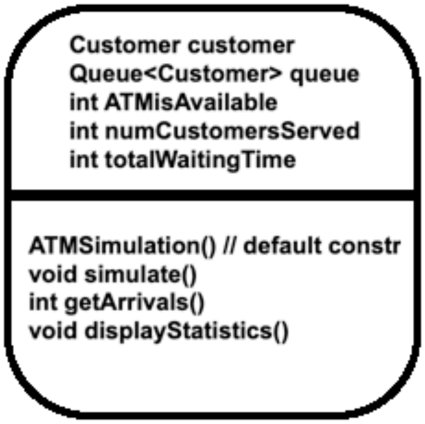
```
For each minute from 0 through 59
{
        Determine the number of new customers arriving:  0, 1, or  2;
        For each new customer
                Place the new customer in the queue;

        If there are customers waiting and the ATM is available
        {
                Remove  a customer from the queue;
                // Update the statistics
                   Increment the number of customers served;
                   Add to the total waiting time the waiting time of the current customer;
                   Update the time the ATM is next available;
        }
}
Print the summary statistics;
```

The program is ATMSimulation.java

The program has three methods:

- void simulate()   // does all the work
- int getArrivals()   // returns the number of arrivals 0,1, or 3
- void displayStatistics()   // prints the summary statistics at the end of the simulation

Here is the basic structure of the class:



class: ATMSimulation

```
import java.util.*;
public class ATMSimulation
{
    private Customer customer;
    private int ATMisAvailable;           // next time the ATM is next available
    private Queue<Customer> queue;

    // statistics
    int totalWaitingTime;                  // for all customers
    int numCustomersServed;

    public ATMSimulation()                 // default constructor
    {
        ATMisAvailable = 0;                // assume the ATM is available at time 0
        totalWaitingTime = 0;
        numCustomersServed = 0;
        queue = new Queue<Customer>(200);
    }




    public  int getArrivals()
                // generate a random integer in the range 0..9
                //  if the random integer is 0,1,2,3,or 4, then no arrivals ( 50% chance)
                //  if the random integer is 5,6,7, or 8, then 1 arrival (40 % chance)
                 //  if the random integer is 9,  then 2 arrivals (10% chance)
    {
        Random rand = new Random();
        int randomInteger = rand.nextInt(10); // 0..9
        if ( randomInteger <= 4) // 0..4
            return 0;  // 50% chance of a single arrival
        if ( randomInteger <= 8) // 5..8
            return 1;  // 40% chance of a single arrival
        return 2; // 10% chance of 2 arrivals
    }




    public void displayStatistics()
    {
        System.out.println("Number of customers served "+ numCustomersServed);
        System.out.println("Average wait is about "+
                              totalWaitingTime/numCustomersServed + "          minutes");
        System.out.println("Customers left in queue: "+ queue.size());
    }
    public void simulate()
    {
        for (int time = 0; time < 60; time++)                   // for each minute
        {
            Int numArrivals = getArrivals();                     // how many customers arrive?
            for (int i = 1; i <= numArrivals; i++)              // place each arrival into the queue
                queue.insert( new Customer(time));

            if (!queue.empty() && ATMisAvailable <= time) //  notice check for empty queue
            {
                customer = queue.remove();      // remove the next customer from the line

                // Determine the next time that the ATM is available: current time+ service time
                ATMisAvailable = time + customer.getServiceTime();

                 // how long did this customer wait?
                int timeCustomerWaited = time - customer.getArrivalTime();

                totalWaitingTime += timeCustomerWaited;   // add customer's wait to total
                numCustomersServed++;
            }
        }
        displayStatistics();
    }

    public static void main(String[] args)
    {
        ATMSimulation atmSim = new ATMSimulation();
        atmSim.simulate();
    }
}
```

Output

Running the application three times produced the following output:

```
Number of customers served  30
Average wait is about 5 minutes
Customers left in queue: 16

Number of customers served 29
Average wait is about 8 minutes
Customers left in queue: 13

Number of customers served 32
Average wait is about 6 minutes
Customers left in queue: 6
```