Simulation Engineering Exercises 03

Name: Chenfeng Zhu
Matrikelnummer: 450485
University: TU-Clausthal
Program: ITIS
Language: JAVA

1. Ice Cream Shop Simulation – Single Server Single Queue

Assumptions:

a) Shops open at 11 AM

b) A new customer arrives randomly between every
1 to 3 minutes till 1 PM
2 to 10 minutes from 1 PM to 5PM
1 to 2 minutes from 5 PM to 7PM
2 to 5 minutes from 7 PM to 8PM

c) Customers ask randomly for
1 to 5 scoops of ice till noon
3 to 6 scoops of ice from noon to 3 PM
1 to 4 scoops of ice from 3 PM to 8 PM

d) Each ice takes 30 seconds

Mathematical Model:

a) Random: $\mathrm{R(n{\sim}m) = Random(0{\sim}(m-n)) + n\ (n < m)}$

b) Expectation:

$$\mathrm{E(X)} = \sum_{k=n}^{m} x_k p_k$$

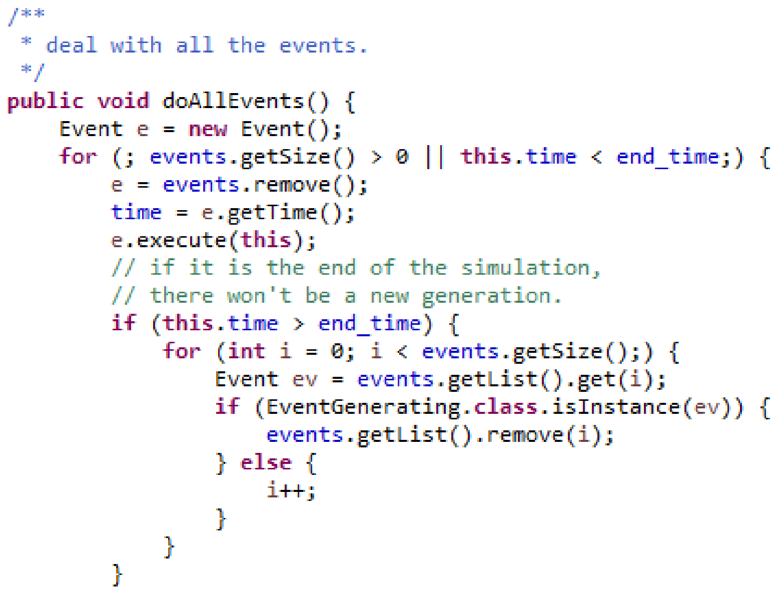
Expectation of Customers: E = $\sum_{k=1}^{3}(\frac{120}{k}\times\frac{1}{3}) + \sum_{k=2}^{10}\left(\frac{240}{k}\times\frac{1}{9}\right) + \sum_{k=1}^{2}\left(\frac{120}{k}\times\frac{1}{2}\right) + \sum_{k=2}^{5}(\frac{60}{k}\times\frac{1}{4})$ = (120/1+120/2+120/3)/3+51.44+90+19.25=234.02

Random Test:

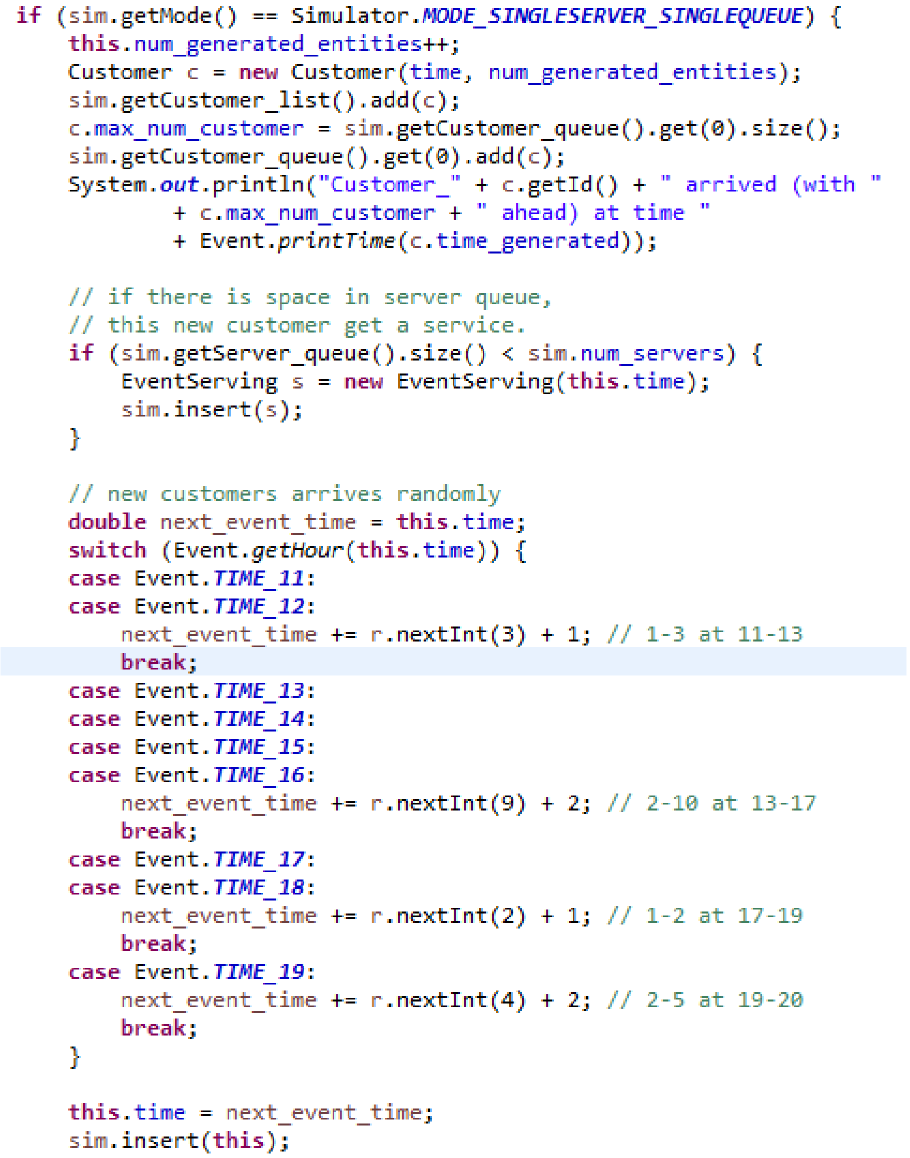
Using *Random.nextInt()* to create a random number. The below is a test:

```
{1=3269, 2=3378, 3=3353}
{2=1076, 3=1114, 4=1094, 5=1130, 6=1112, 7=1162, 8=1102, 9=1105, 10=1105}
{1=4980, 2=5020}
{2=2468, 3=2529, 4=2523, 5=2480}
```

Main code:



```
/**
 * deal with all the events.
 */
public void doAllEvents() {
    Event e = new Event();
    for (; events.getSize() > 0 || this.time < end_time;) {
        e = events.remove();
        time = e.getTime();
        e.execute(this);
        // if it is the end of the simulation,
        // there won't be a new generation.
        if (this.time > end_time) {
            for (int i = 0; i < events.getSize();) {
                Event ev = events.getList().get(i);
                if (EventGenerating.class.isInstance(ev)) {
                    events.getList().remove(i);
                } else {
                    i++;
                }
            }
        }
```

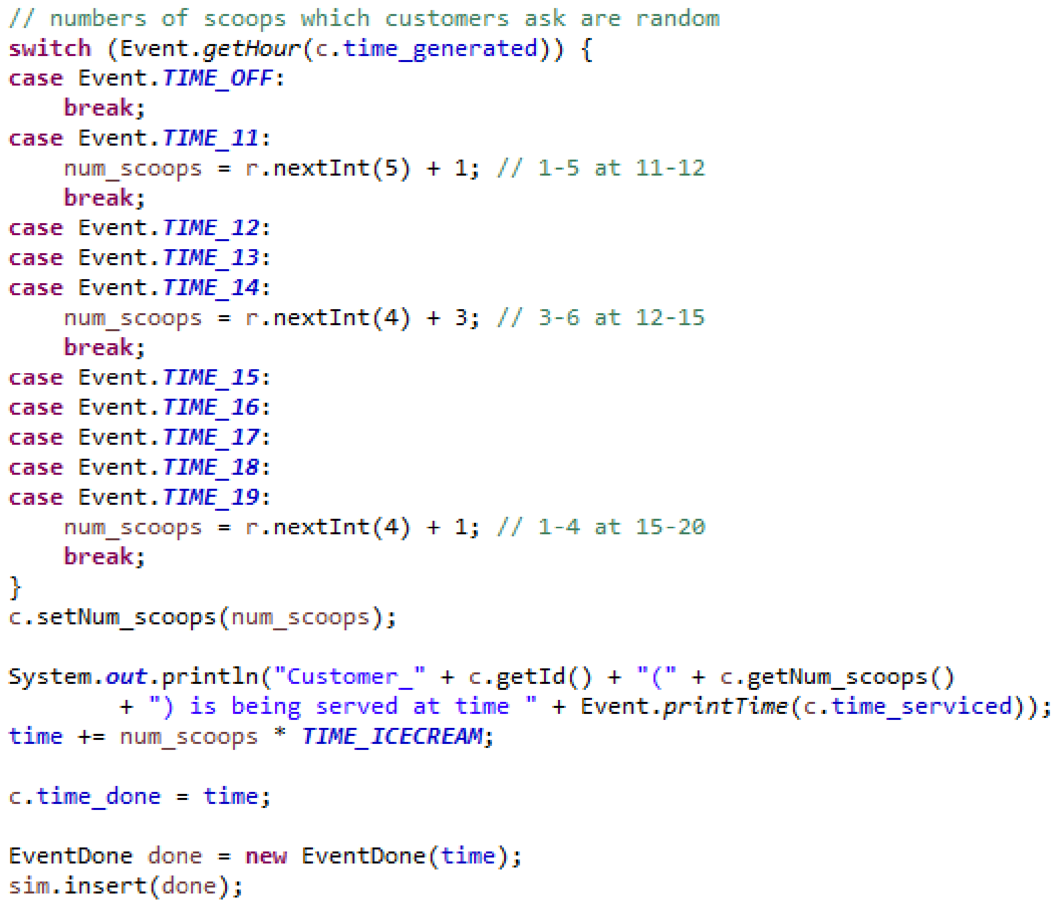


```
if (sim.getMode() == Simulator.MODE_SINGLESERVER_SINGLEQUEUE) {
    this.num_generated_entities++;
    Customer c = new Customer(time, num_generated_entities);
    sim.getCustomer_list().add(c);
    c.max_num_customer = sim.getCustomer_queue().get(0).size();
    sim.getCustomer_queue().get(0).add(c);
    System.out.println("Customer_" + c.getId() + " arrived (with "
            + c.max_num_customer + " ahead) at time "
            + Event.printTime(c.time_generated));

    // if there is space in server queue,
    // this new customer get a service.
    if (sim.getServer_queue().size() < sim.num_servers) {
        EventServing s = new EventServing(this.time);
        sim.insert(s);
    }

    // new customers arrives randomly
    double next_event_time = this.time;
    switch (Event.getHour(this.time)) {
    case Event.TIME_11:
    case Event.TIME_12:
        next_event_time += r.nextInt(3) + 1; // 1-3 at 11-13
        break;
    case Event.TIME_13:
    case Event.TIME_14:
    case Event.TIME_15:
    case Event.TIME_16:
        next_event_time += r.nextInt(9) + 2; // 2-10 at 13-17
        break;
    case Event.TIME_17:
    case Event.TIME_18:
        next_event_time += r.nextInt(2) + 1; // 1-2 at 17-19
        break;
    case Event.TIME_19:
        next_event_time += r.nextInt(4) + 2; // 2-5 at 19-20
        break;
    }

    this.time = next_event_time;
    sim.insert(this);
```



```
// numbers of scoops which customers ask are random
switch (Event.getHour(c.time_generated)) {
case Event.TIME_OFF:
    break;
case Event.TIME_11:
    num_scoops = r.nextInt(5) + 1; // 1-5 at 11-12
    break;
case Event.TIME_12:
case Event.TIME_13:
case Event.TIME_14:
    num_scoops = r.nextInt(4) + 3; // 3-6 at 12-15
    break;
case Event.TIME_15:
case Event.TIME_16:
case Event.TIME_17:
case Event.TIME_18:
case Event.TIME_19:
    num_scoops = r.nextInt(4) + 1; // 1-4 at 15-20
    break;
}
c.setNum_scoops(num_scoops);

System.out.println("Customer_" + c.getId() + "(" + c.getNum_scoops()
        + ") is being served at time " + Event.printTime(c.time_serviced));
time += num_scoops * TIME_ICECREAM;

c.time_done = time;

EventDone done = new EventDone(time);
sim.insert(done);
```

Graph:

NULL.

Output results:

Test the simulation 3 times.

a) IceCreamShopSimulation_SSSQ01:

Time: 0.0-1440.0 (1 day)

Customer Quantity: 192

Total scoops made: 595

Average scoops per customer: 3.0989583333333335

The customer waiting longest: Customer_57 with 4 scoops (arrived at 765.0(Day_0 12h:45.0m) with 4 customers ahead, was serviced at 772.0(Day_0 12h:52.0m), left at 774.0(Day_0 12h:54.0m))

Total waiting time: 420.0

Average waiting time per customer: 2.1875

Customers at different periods: {1=33, 2=30, 3=9, 4=9, 5=10, 6=10, 7=34, 8=39, 9=18}

Total scoops at different periods: {1=97.0, 2=122.0, 3=46.0, 4=46.0, 5=33.0, 6=22.0, 7=84.0, 8=101.0, 9=44.0}

Total waiting time at different periods: {1=68.5, 2=131.0, 3=26.0, 4=23.0, 5=16.5, 6=11.0, 7=63.0, 8=58.5, 9=22.5}

Average scoops at different periods: {1=2.939, 2=4.066, 3=5.111, 4=5.111, 5=3.3, 6=2.2, 7=2.47, 8=2.589, 9=2.444}

Average waiting time at different periods: {1=2.075, 2=4.366, 3=2.888, 4=2.555, 5=1.65, 6=1.1, 7=1.852, 8=1.5, 9=1.25}

b) IceCreamShopSimulation_SSSQ02:

Time: 0.0-5880.0 (4 days)

Customer Quantity: 791

Total scoops made: 2427

Average scoops per customer: 3.06826801517067

The customer waiting longest: Customer_262 with 4 scoops (arrived at 2215.0(Day_1 12h:55.0m) with 8 customers ahead, was serviced at 2229.0(Day_1 13h:9.0m), left at 2231.0(Day_1 13h:11.0m))

Total waiting time: 2339.5

Average waiting time per customer: 2.9576485461441213

Customers at different periods: {1=125, 2=124, 3=43, 4=39, 5=39, 6=38, 7=154, 8=159, 9=70}

Total scoops at different periods: {1=356.0, 2=573.0, 3=180.0, 4=189.0, 5=93.0, 6=98.0, 7=397.0, 8=375.0, 9=166.0}

Total waiting time at different periods: {1=244.0, 2=1001.0, 3=230.5, 4=97.5, 5=46.5, 6=49.0, 7=285.5, 8=301.0, 9=84.5}

Average scoops at different periods: {1=2.848, 2=4.62, 3=4.186, 4=4.846, 5=2.384, 6=2.578, 7=2.577, 8=2.358, 9=2.371}

Average waiting time at different periods: {1=1.952, 2=8.072, 3=5.36, 4=2.5, 5=1.192, 6=1.289, 7=1.853, 8=1.893, 9=1.207}

c) IceCreamShopSimulation_SSSQ03:

Time: 0.0-14400.0 (10 days)

Customer Quantity: 1976

Total scoops made: 6147

Average scoops per customer: 3.11082995951417

The customer waiting longest: Customer_49 with 4 scoops (arrived at 749.0(Day_0 12h:29.0m) with 9 customers ahead, was serviced at 767.0(Day_0 12h:47.0m), left at 769.0(Day_0 12h:49.0m))

Total waiting time: 5771.5

Average waiting time per customer: 2.9207995951417005

Customers at different periods: {1=291, 2=303, 3=100, 4=112, 5=97, 6=108, 7=386, 8=398, 9=181}

Total scoops at different periods: {1=885.0, 2=1382.0, 3=464.0, 4=488.0, 5=225.0, 6=265.0, 7=936.0, 8=1032.0, 9=470.0}

Total waiting time at different periods: {1=570.5, 2=2416.5, 3=431.5, 4=253.5, 5=112.5, 6=132.5, 7=656.5, 8=956.0, 9=242.0}

Average scoops at different periods: {1=3.041, 2=4.561, 3=4.64, 4=4.357, 5=2.319, 6=2.453, 7=2.424, 8=2.592, 9=2.596}

Average waiting time at different periods: {1=1.96, 2=7.975, 3=4.315, 4=2.263, 5=1.159, 6=1.226, 7=1.7, 8=2.402, 9=1.337}

Conclusion:

According to the result, there would be around 220 customers to serve, about 600 scoops to make in one day.

In one day, there would be most customers to serve from 17:00 to 19:00 and most scoops to make from 12:00 to 13:00.

The busiest time for Mike is 12:00 to 13:00, and the customers who arrived at that time have to wait for a longest time.

Attention: even though the customers arrived from 13:00 to 14:00 have to wait for a longer time, it does not mean that there are more customers and scoops. The reason is that there are too many scoops left to make in the last hour.

Answers:

a) How the queue changes over the day?

The quantity of customers keeps steady during 11:00 to 13:00. And it goes down from 13:00 to 17:00. Then it rises to the top from 17:00 to 19:00. The customers become few in the last hour (19:00 to 20:00).

According to the average waiting time, the queue keeps smooth at most time except when it is from 12:00 and this busy situation keeps for 2 hours until 14:00.

b) Does Mike needs to employ somebody else?

Yes. I think Mike needs to employ somebody else. As to how many would be OK, we should simulation other situations.

c) Full time or Part time

Part time during the busiest time (from 12:00 to 13:00).

2. Ice Cream Shop Simulation – Multiple Servers Single Queue

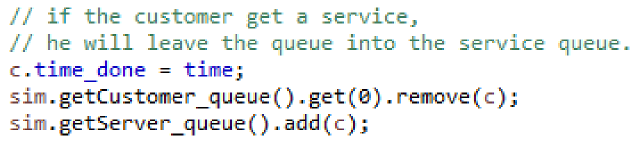
Assumptions:

It is the same as the simple one. But there would be two servers.

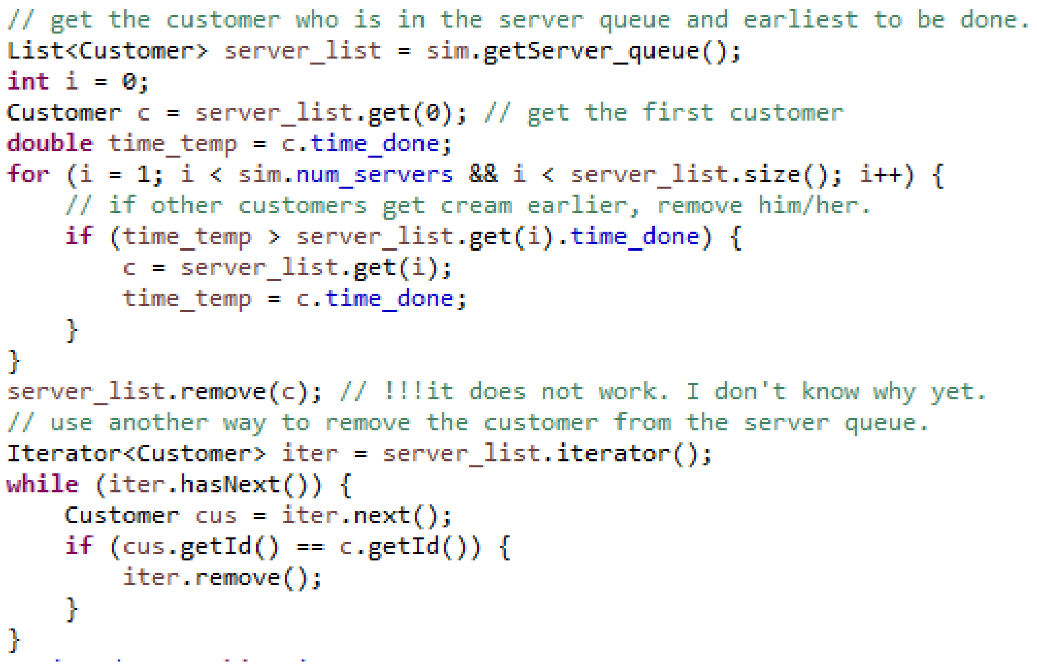
Solution:

Improve the code of the first simulation. Create a server queue. The creation of the Generating Event depends on both the customer queue and server queue. When the event of Generating is created, move the first customer in the customer queue to the server queue. When the event of Done is created, remove the customer who is in the server queue and has the earliest time of done.

Main code:



```
// if the customer get a service,
// he will leave the queue into the service queue.
c.time_done = time;
sim.getCustomer_queue().get(0).remove(c);
sim.getServer_queue().add(c);
```



```
// get the customer who is in the server queue and earliest to be done.
List<Customer> server_list = sim.getServer_queue();
int i = 0;
Customer c = server_list.get(0); // get the first customer
double time_temp = c.time_done;
for (i = 1; i < sim.num_servers && i < server_list.size(); i++) {
    // if other customers get cream earlier, remove him/her.
    if (time_temp > server_list.get(i).time_done) {
        c = server_list.get(i);
        time_temp = c.time_done;
    }
}
server_list.remove(c); // !!!it does not work. I don't know why yet.
// use another way to remove the customer from the server queue.
Iterator<Customer> iter = server_list.iterator();
while (iter.hasNext()) {
    Customer cus = iter.next();
    if (cus.getId() == c.getId()) {
        iter.remove();
    }
}
```

Graph:

NULL.

Output results:

Test the simulation 3 times.

a) IceCreamShopSimulation_MSSQ01:

Time: 0.0-1440.0 (1 day)

Customer Quantity: 189

Total scoops made: 586

Average scoops per customer: 3.1005291005291005

The customer waiting longest: Customer_35 with 5 scoops (arrived at 726.0(Day_0 12h:6.0m) with 2 customers ahead, was serviced at 726.5(Day_0 12h:6.5m), left at 729.0(Day_0 12h:9.0m))

Total waiting time: 296.0

Average waiting time per customer: 1.566137566137566

Customers at different periods: {1=31, 2=32, 3=11, 4=9, 5=10, 6=8, 7=35, 8=35, 9=18}

Total scoops at different periods: {1=89.0, 2=148.0, 3=49.0, 4=42.0, 5=25.0, 6=18.0, 7=93.0, 8=86.0, 9=36.0}

Total waiting time at different periods: {1=44.5, 2=77.0, 3=24.5, 4=21.0, 5=12.5, 6=9.0, 7=46.5, 8=43.0, 9=18.0}

Average scoops at different periods: {1=2.87, 2=4.625, 3=4.454, 4=4.666, 5=2.5, 6=2.25, 7=2.657, 8=2.457, 9=2.0}

Average waiting time at different periods: {1=1.435, 2=2.406, 3=2.227, 4=2.333, 5=1.25, 6=1.125, 7=1.328, 8=1.228, 9=1.0}

b) IceCreamShopSimulation_MSSQ02:

Time: 0.0-7200.0 (5 days)

```
Customer Quantity: 994
Total scoops made: 3055
Average scoops per customer: 3.073440643863179
The customer waiting longest: Customer_652 with 5 scoops (arrived at 5090.0(Day_3
12h:50.0m) with 3 customers ahead, was serviced at 5090.0(Day_3 12h:50.0m), left at
5092.5(Day_3 12h:52.5m))
Total waiting time: 1535.5
Average waiting time per customer: 1.54476861167002
Customers at different periods: {1=149, 2=152, 3=55, 4=52, 5=55, 6=50, 7=192, 8=201, 9=88}
Total scoops at different periods: {1=449.0, 2=691.0, 3=226.0, 4=234.0, 5=137.0, 6=121.0,
7=452.0, 8=510.0, 9=235.0}
Total waiting time at different periods: {1=226.5, 2=351.0, 3=113.5, 4=117.0, 5=68.5, 6=60.5,
7=226.0, 8=255.0, 9=117.5}
Average scoops at different periods: {1=3.013, 2=4.546, 3=4.109, 4=4.5, 5=2.49, 6=2.42,
7=2.354, 8=2.537, 9=2.67}
Average waiting time at different periods: {1=1.52, 2=2.309, 3=2.063, 4=2.25, 5=1.245,
6=1.21, 7=1.177, 8=1.268, 9=1.335}
```

c) IceCreamShopSimulation_MSSQ03:

```
Time: 0.0-14400.0 (10 days)
Customer Quantity: 1986
Total scoops made: 6171
Average scoops per customer: 3.107250755287009
The customer waiting longest: Customer_34 with 3 scoops (arrived at 726.0(Day_0 12h:6.0m)
with 2 customers ahead, was serviced at 726.5(Day_0 12h:6.5m), left at 728.0(Day_0
12h:8.0m))
Total waiting time: 3098.0
Average waiting time per customer: 1.5599194360523665
Customers at different periods: {1=301, 2=297, 3=104, 4=100, 5=106, 6=102, 7=388, 8=408,
9=180}
Total scoops at different periods: {1=956.0, 2=1355.0, 3=456.0, 4=472.0, 5=259.0, 6=249.0,
7=998.0, 8=968.0, 9=458.0}
Total waiting time at different periods: {1=480.5, 2=687.5, 3=228.0, 4=236.0, 5=129.5,
6=124.5, 7=499.0, 8=484.0, 9=229.0}
Average scoops at different periods: {1=3.176, 2=4.562, 3=4.384, 4=4.72, 5=2.443, 6=2.441,
7=2.572, 8=2.372, 9=2.544}
Average waiting time at different periods: {1=1.596, 2=2.314, 3=2.192, 4=2.36, 5=1.221,
6=1.22, 7=1.286, 8=1.186, 9=1.272}
```

Conclusion:

From the result of the second simulation, we could find that the waiting time of the customers who arrive during 12:00 to 14:00 has been shorted.

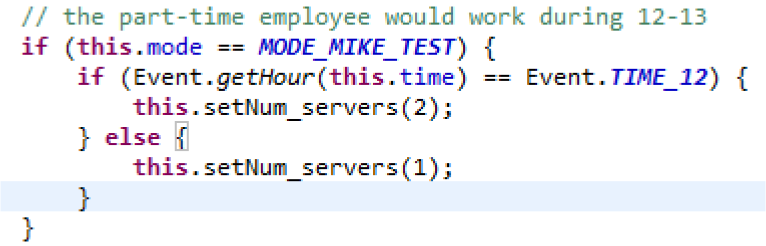
However, the waiting time of the customers who arrive at other time does not change too much. That means that the other servers are free at most time.

3. Ice Cream Shop Simulation – Mike hires a part-time employee

Assumptions:

It is the same as the simple one.
But there would be two servers during 12:00 to 13:00.

Main code:



```
// the part-time employee would work during 12-13
if (this.mode == MODE_MIKE_TEST) {
    if (Event.getHour(this.time) == Event.TIME_12) {
        this.setNum_servers(2);
    } else {
        this.setNum_servers(1);
    }
}
```

Graph:

NULL.

Output results:

Test the simulation 3 times.

a) IceCreamShopSimulation_TEST01:

Time: 0.0-1440.0 (1 days)

Customer Quantity: 195

Total scoops made: 617

Average scoops per customer: 3.164102564102564

The customer waiting longest: Customer_119 with 4 scoops (arrived at 1048.0(Day_0 17h:28.0m) with 5 customers ahead, was serviced at 1053.5(Day_0 17h:33.5m), left at 1055.5(Day_0 17h:35.5m))

Total waiting time: 466.5

Average waiting time per customer: 2.3923076923076922

Customers at different periods: {1=31, 2=28, 3=11, 4=10, 5=9, 6=9, 7=42, 8=37, 9=18}

Total scoops at different periods: {1=95.0, 2=124.0, 3=44.0, 4=44.0, 5=25.0, 6=25.0, 7=115.0, 8=93.0, 9=52.0}

Total waiting time at different periods: {1=73.5, 2=63.0, 3=22.5, 4=22.5, 5=12.5, 6=12.5, 7=174.5, 8=58.5, 9=27.0}

Average scoops at different periods: {1=3.064, 2=4.428, 3=4.0, 4=4.4, 5=2.777, 6=2.777, 7=2.738, 8=2.513, 9=2.888}

Average waiting time at different periods: {1=2.37, 2=2.25, 3=2.045, 4=2.25, 5=1.388, 6=1.388, 7=4.154, 8=1.581, 9=1.5}

b) IceCreamShopSimulation_TEST02:

Time: 0.0-7200.0 (5 days)

Customer Quantity: 972

Total scoops made: 3031

Average scoops per customer: 3.1183127572016462

The customer waiting longest: Customer_322 with 2 scoops (arrived at 2505.0(Day_1 17h:45.0m) with 3 customers ahead, was serviced at 2507.5(Day_1 17h:47.5m), left at 2508.5(Day_1 17h:48.5m))

Total waiting time: 1797.0

Average waiting time per customer: 1.8487654320987654

Customers at different periods: {1=147, 2=154, 3=53, 4=52, 5=54, 6=45, 7=190, 8=192, 9=85}

Total scoops at different periods: {1=430.0, 2=692.0, 3=231.0, 4=247.0, 5=149.0, 6=102.0, 7=481.0, 8=475.0, 9=224.0}

Total waiting time at different periods: {1=274.0, 2=352.0, 3=116.5, 4=126.5, 5=74.5, 6=51.0, 7=360.0, 8=329.0, 9=113.5}

Average scoops at different periods: {1=2.925, 2=4.493, 3=4.358, 4=4.75, 5=2.759, 6=2.266, 7=2.531, 8=2.473, 9=2.635}

Average waiting time at different periods: {1=1.863, 2=2.285, 3=2.198, 4=2.432, 5=1.379, 6=1.133, 7=1.894, 8=1.713, 9=1.335}

c) IceCreamShopSimulation_TEST03:

Time: 0.0-14400.0 (10 days)

Customer Quantity: 1987

Total scoops made: 6151

Average scoops per customer: 3.0956215400100655

The customer waiting longest: Customer_825 with 1 scoops (arrived at 6466.0(Day_4 11h:46.0m) with 6 customers ahead, was serviced at 6473.0(Day_4 11h:53.0m), left at 6473.5(Day_4 11h:53.5m))

Total waiting time: 3855.5

Average waiting time per customer: 1.9403623553095117

Customers at different periods: {1=306, 2=310, 3=109, 4=98, 5=104, 6=102, 7=385, 8=397, 9=176}

Total scoops at different periods: {1=895.0, 2=1388.0, 3=491.0, 4=419.0, 5=266.0, 6=256.0, 7=974.0, 8=1019.0, 9=443.0}

Total waiting time at different periods: {1=684.0, 2=710.5, 3=250.0, 4=214.0, 5=133.0, 6=128.0, 7=730.5, 8=780.0, 9=225.5}

Average scoops at different periods: {1=2.924, 2=4.477, 3=4.504, 4=4.275, 5=2.557, 6=2.509, 7=2.529, 8=2.566, 9=2.517}

Average waiting time at different periods: {1=2.235, 2=2.291, 3=2.293, 4=2.183, 5=1.278, 6=1.254, 7=1.897, 8=1.964, 9=1.281}

Conclusion:

The result is satisfied.

Code:

https://github.com/sampig/SimulationEngineering