

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Э. БАУМАНА

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»



---

# Моделирование

---

ЛАБОРАТОРНАЯ РАБОТА №6  
МОДЕЛИРОВАНИЕ КАСС В СУПЕРМАРКЕТЕ

Студент: Петухов И.С.

Группа: ИУ7-71

Преподаватель: Рудаков И.В.

Москва, 2016

## Содержание

1	Аналитический раздел . . . . .	3
2	Технологический раздел . . . . .	5
2.1	Язык программирования . . . . .	5
2.2	Примеры кода . . . . .	5
2.3	Взаимодействие с пользователем . . . . .	7

## 1 Аналитический раздел

**Цель** данной работы - смоделировать работу касс супермаркета.

К кассам супермаркета идет поток покупателей. Данный поток - Пуассоновский. Параметр задается.

Каждый покупатель держит несколько товаров. Количество товаров у пользователя распределено равномерно.

Есть два типа касс:

- а) быстрые
- б) обычные

Быстрые кассы обслуживают только тех клиентов, у кого количество товаров меньше  $N$ .  $N$  - задается. Количество касс задается.

Обычные кассы обслуживают остальных пользователей. Количество касс задается.

Время обработки кассиром одного товара константное, задается. Время работы кассира равно времени обработки одного товара умноженное на количество товаров у покупателя.

Вывести количество пользователей в очереди к каждой кассе в каждый момент времени при заданном количестве покупателей, которых должен обслужить супермаркет.

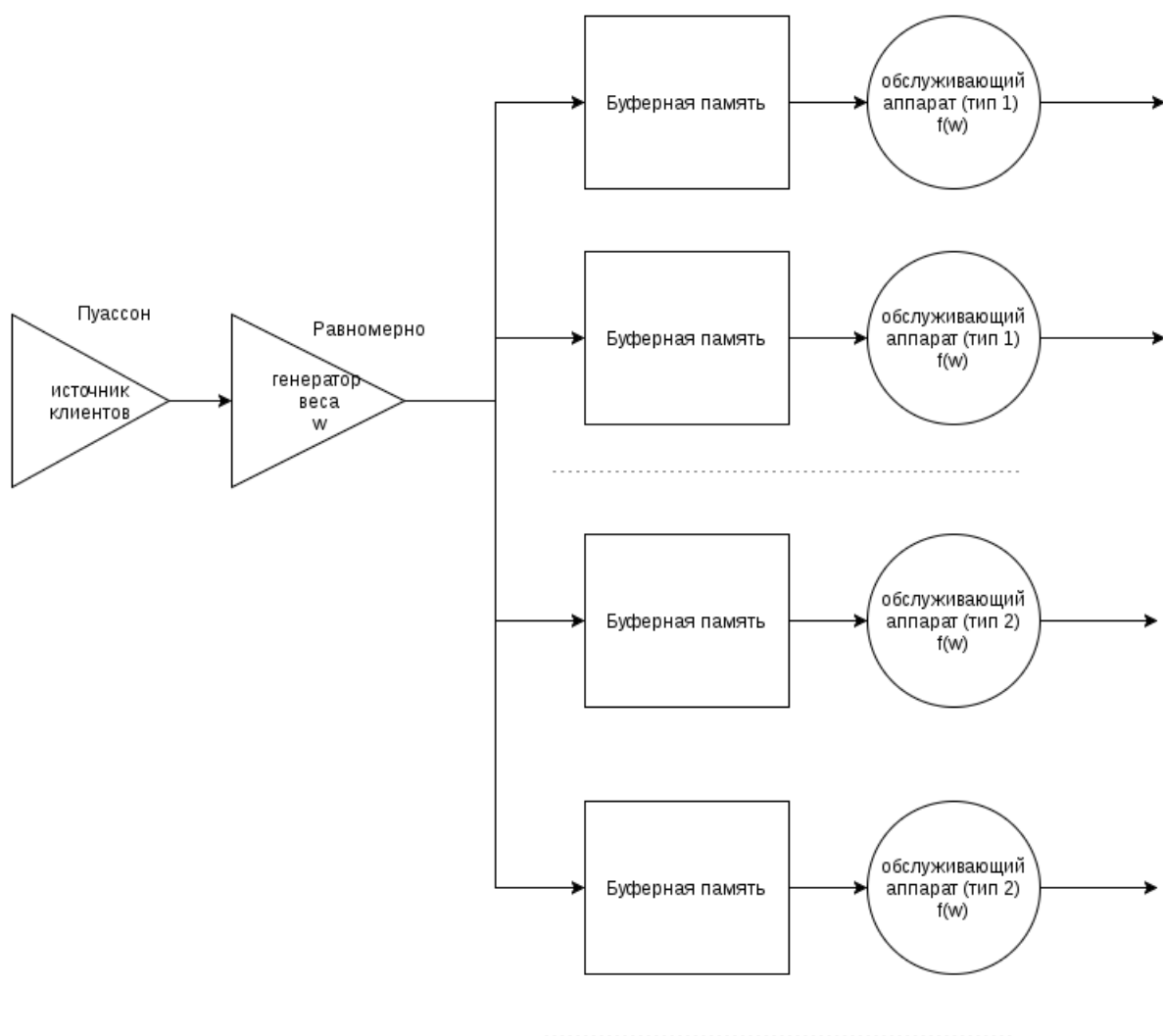


Рисунок 1.1 — Диаграмма модели

## 2 Технологический раздел

### 2.1 Язык программирования

В качестве языка программирования выбран язык высокого уровня JavaScript.

### 2.2 Примеры кода

Листинг 2.1 — Реализация имитационной модели

```
1  while (amountOutput < aBuyer.amount) {
2
3      if (buyer.isRequest(nowTime)) {
4
5          let numberCashier = 0;
6          let pollCashiers;
7
8          if (buyer.getWeightReq() <= cashier_fast.max_weight) {
9              pollCashiers = fastCashiers;
10         } else {
11             pollCashiers = cashiers;
12         }
13
14         pollCashiers.forEach(function(item, i, arr) {
15             if (item.getLengthQueue() < arr[numberCashier].getLengthQueue()) {
16                 numberCashier = i;
17             }
18         })
19         pollCashiers[numberCashier].pushWork(nowTime, buyer.getWeightReq())
20
21
22         buyer.reset();
23
24         ++amountInput;
25     }
26
27     fastCashiers.forEach(function(el) {
28         amountOutput += el.check(nowTime);
29     })
30
31     cashiers.forEach(function(el) {
32         amountOutput += el.check(nowTime);
33     })
34
35     nowTime += simulator.dt;
36 }
```

## Листинг 2.2 — Программная модель кассы

```

1 function ServiceUnit(timePerRequest) {
2   let self = this;
3
4   self.queue = [];
5   self.tpr = timePerRequest;
6
7   let timeNewWork;
8
9   self.getLengthQueue = function() {
10    return self.queue.length;
11  }
12
13  self.check = function(nowTime) {
14    if (self.queue.length == 0) return 0;
15
16    let finished
17
18    if (nowTime >= timeNewWork) {
19      finished = self.queue.shift()
20    }
21
22    if (finished) {
23      timeNewWork = nowTime + self.queue[0] * self.tpr;
24      return 1
25    }
26
27    return 0;
28  }
29
30  self.pushWork = function(nowTime, weight) {
31    if (self.queue.length == 0)
32      timeNewWork = nowTime + weight * self.tpr;
33
34    self.queue.push(weight)
35  }
36 }

```

## Листинг 2.3 — Программная модель покупателя

```

1 function SourceOfInformation(timeReq, weightReq) {
2   let self = this;
3
4   self.timeReq = timeReq;
5   self.weightReq = weightReq;
6
7   let isRequest = false;

```

```

8
9   let timeNewRequest = self.timeReq.randFunc(self.timeReq.param);
10
11  let weightNewRequest = self.weightReq.randFunc(self.weightReq.param);
12
13  self.isRequest = function(nowTime) {
14      if (nowTime >= timeNewRequest && !isRequest) {
15          isRequest = true;
16          timeNewRequest += self.timeReq.randFunc(self.timeReq.param);
17          weightNewRequest = self.weightReq.randFunc(self.weightReq.param);
18      }
19
20      return isRequest;
21  }
22
23  self.getWeightReq = function() {
24      return weightNewRequest;
25  }
26
27  self.reset = function() {
28      isRequest = false;
29  }
30 }

```

## 2.3 Взаимодействие с пользователем

Взаимодействие с пользователем осуществляется через html страницы, открытые в браузере.

---

## Управляющая программа имитационной модели супермаркета (принцип dt)

### Параметры имитационной модели

шаг времени:

---

### Покупатели

кол-во обслуженных:

параметр  $\lambda$  Пуассоновского потока:

### количество товаров у покупателя:

min:

max:

---

### Быстрые кассы:

кол-во:

максимальное число товаров:

---

### Обычные кассы:

кол-во:

время обработки одного товара:

---

**Запуск**

Рисунок 2.1 — Настройка параметров модели



Состояние очередей на быстрые кассы

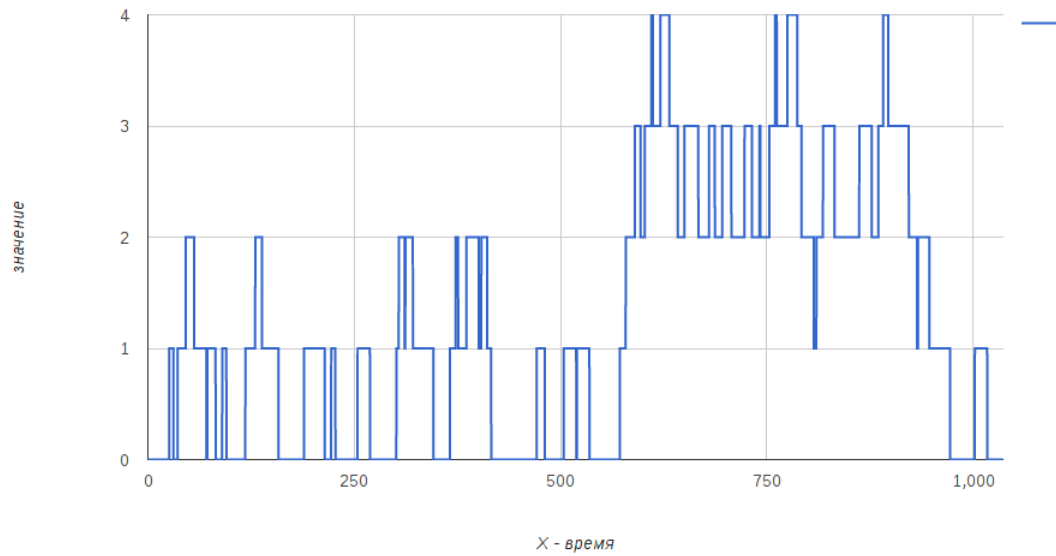


Рисунок 2.2 — Длина очереди в каждой быстрой кассе

Состояние очередей на обычные кассы

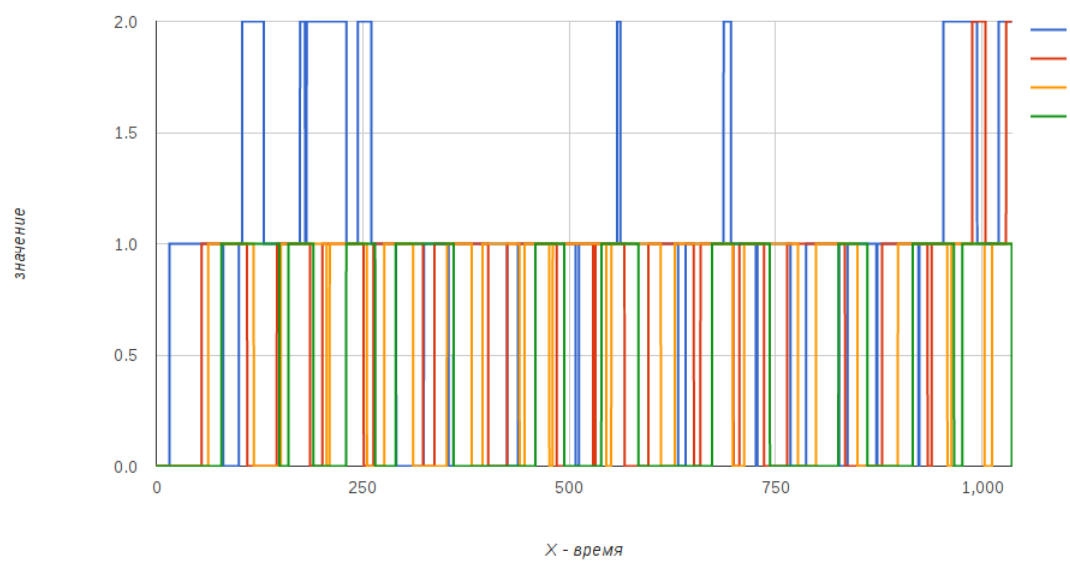


Рисунок 2.3 — Длина очереди в каждой обычной кассе