Федеральное агентство по образованию

ГОУ ВПО «Нижегородский государственный архитектурно – строительный университет»

Международный институт экономики, права и менеджмента

Кафедра информационных систем в экономике

Курсовая работа

по дисциплине: Математические методы и модели

на тему:

«Имитационное моделирование группового обслуживания с несколькими этапами и двойной очередью: работа оптового магазина»

Выполнил студент:

Чикунова Е. О.

Проверил:

Прокопенко Н. Ю.

г. Н. Новгород

2010 г.

**Оглавление**

Введение

1 Имитационное моделирование

2 Описание системы

2.1 Модельное время

2.2 Классы и объекты

2.3 События и методы

3 Реализация модели

3.1 Программная реализация

3.2 Построение графиков

3.2.1 Программа gnuplot

3.2.2 Использование программы для построения графиков

4 Анализ результатов

Список использованной литературы

**Введение**

В современном мире гарантией эффективной работы любого предприятия служит рациональное использование денежных средств и трудового фактора. Именно поэтому для расчета экономического эффекта работы оптового магазина необходимо провести имитационное моделирование на основании предварительно установленных зависимостей.

Термин имитационное моделирование означает, что речь идет о моделях с помощью, которых нельзя вычислить или предсказать результат и поэтому с их помощью проводиться вычислительный эксперимент при заданных исходных данных.

Метод имитационного моделирования дает возможность широкого использования математического аппарата и вычислительной техники для исследования хода экономических процессов.

Таким образом, сущность имитационного моделирования состоит в том, что с помощью ЭВМ воспроизводится поведение исследуемой системы, а исследователь, управляет ходом процесса и анализирует получаемые результаты. Поэтому под имитацией следует понимать численный метод проведения на ЭВМ экспериментов с алгоритмами, описывающими поведение системы и определения интересующих нас функциональных характеристик.

Целью данной курсовой работы является разработка модели группового обслуживания с несколькими этапами и двойной очередью, то есть работа оптового магазина. Основой для разработки модели в данной курсовой работе является метод имитационного моделирования. Так же курсовая работа предполагает создание программы на языке C++, обеспечивающей ввод исходной информации, ее обработку, реализацию алгоритма имитации процесса и выдачу необходимой информации.

**1. Имитационное моделирование**

Можно дать следующее определение понятия *модель:* это такое описание, которое исключает несущественные подробности и учитывает наиболее важные особенности системы. Моделирование же можно определить как методологию изучения системы путем наблюдения отклика модели на искусственно генерируемый входной поток. К. Шеннон пишет так: «Имитационное моделирование есть процесс конструирования модели реальной системы и постановки экспериментов на этой модели с целью либо понять поведение системы, либо оценить (в рамках ограничений, накладываемых некоторым критерием или совокупностью критериев) различные стратегии, обеспечивающие функционирование данной системы...» Имитационное моделирование является экспериментальной и прикладной методологией, имеющей следующие цели [1]:

* Описание поведения системы;
* Построение теорий и гипотез, которые могут объяснить наблюдаемое поведение;
* Использование этих теорий для предсказания будущего поведения системы, то есть тех воздействий, которые могут быть вызваны изменениями в системе или изменениями способов ее функционирования.

Авторы одной методологической работы сформулировали основные факторы, влияющие на принятие правильного решения по результатам моделирования:

* адекватное понимание решаемой задачи,т. е. если задача не полностью определена и недостаточно четко описана, очень мало шансов, что ее решение принесет какую-либо пользу. Это фундаментальное утверждение относится ко всем задачам, а не только к моделированию.
* корректная модель*.* Это первостепенный фактор для технически или экономически эффективного решения, если брать всю задачу в целом. Ошибки в модели, если они не выявлены, скорее всего, приведут к принятию результатов, основанных на неверной модели. Стоимость такого типа ошибок обычно очень высока. Даже если ошибка обнаружена, но это произошло на поздних этапах проекта, стоимость исправлений включает также и повторное прохождение всех предшествующих этапов.
* корректная программа*.* Программирование — последний этап разработки, и корректная программа может быть написана только по корректной модели. Аргументы в пользу корректности программы такие же, что и для модели.
* планирование эксперимента*.* Разработка модели и программы должна отражать цели, для которых выполняется моделирование. Для получения требуемых ответов программе нужно правильно задать вопросы, то есть спланировать последовательность вычислительных экспериментов с полным пониманием проблемы.
* интерпретация результатов*.* Никакая моделирующая программа не дает ответа со стопроцентной достоверностью. Результаты моделирования получаются на основе обработки случайных чисел, поэтому для их правильного понимания требуется применение статистических методов.

Таким образом, моделирование — это больше, чем просто программа. Достижение целей моделирования требует пристального внимания ко всем указанным факторам.

Типовая последовательность имитационного моделирования включает следующие этапы [1]:

1. *Концептуальный:* разработка концептуальной схемы и подготовка области исходных данных;
2. *Математический:* разработка математических моделей и обоснование методов моделирования;
3. *Программный:* выбор средств моделирования и разработка программных моделей;
4. *Экспериментальный:* проверка адекватности и корректировка моделей, планирование вычислительных экспериментов, непосредственно моделирование, интерпретация результатов.

Имитационное моделирование на компьютере, в принципе, позволяет проанализировать любую реальную систему произвольной сложности. Концептуально, промоделировать сложную систему так же легко, как и простую, разница будет состоять только в объеме программного кода. Имитационная модель может учесть любой нюанс в дисциплине обслуживания всего лишь путем небольшой модификации текста одной-двух процедур, а в аналитической модели это может потребовать коренной переделки всех уравнений, сделать модель необозримо сложной или оказаться вообще невозможным. Этот факт отражает как силу, так и слабость имитационной методологии. С одной стороны, имитационное моделирование даст метод анализа, применимый в тех случаях, когда математическая модель чрезмерно сложна и позволяет аналитику получить более точные результаты. Но с другой стороны, имитационная модель не позволяет глубоко заглянуть в сущность системы, выявить ее «изюминки» и законы, по которым она живет, построить качественные зависимости между «входом» и «выходом», как это позволяет сделать математическая модель, если ее, конечно, удалось решить. То, что при взгляде на математический результат видно сразу, при имитационном моделировании может быть выявлено только в результате постановки значительного количества экспериментов (еще говорят «прогонов»)[1].

Главная и наиболее очевидная цель имитационного моделирования — выяснить, как повлияют на производительность отдельные изменения конфигурации системы или увеличение нагрузки на нее. Процесс моделирования включает три фазы. На фазе *валидации* строится базовая модель существующей системы, проверяются и обосновываются предположения, лежащие в ее основе. На фазе *проектирования* модель используется в прогностических целях для предсказания влияния различных модификаций на производительность. На фазе *верификации* реальная производительность модифицированной системы сравнивается с результатами моделирования. Взятые вместе, эти три фазы образуют модельный цикл [1].

Фаза валидации.

Начинается с описания модели и включает выбор тех ресурсов и элементов деятельности, которые будут представлены; выявление особенностей системы, которые требуют внимания; выбор структуры модели; процедуры расчета необходимых показателей по результатам имитационного эксперимента.

Далее в реально функционирующей системе проводятся замеры входных параметров, которые послужат рабочим материалом для модели, а также замеры производительности, результаты которых будут сравниваться с выходными данными модели для оценки ее точности. Модель проверяется, в результате чего может потребоваться внести в нее изменения. Значимые различия между выходными данными системы и модели свидетельствуют об изъянах модели - какое-то допущение оказалось некорректным, какие-то факторы проигнорированы неправомерно. Но и отсутствие таких различий еще не гарантирует того, что модель сумеет правильно предвидеть влияние количественных и качественных изменений в системе.

Фаза проектирования.

На этой фазе входные параметры меняются в соответствии с модификацией системы, эффективность которой нужно проверить с помощью модели. Это довольно сложный и ответственный процесс, ведь необходимо правильно сформулировать вопрос дли модели. Результаты затем анализируются, их отличия от выходных данных исходной модели и представляют собой эффект от модификации системы.

Фаза верификации.

На фазе верификации измерения снимаются с обновленной системы, и снова проводится сравнение. Производительность системы сравнивается с данными моделирования. Наблюдаемые различия могут объясняться двумя причинами:

* либо при составлении модели упущены некоторые ее свойства, что дает о себе знать не всегда, а лишь при стечении определенных обстоятельств;
* либо система отреагировала на изменения совсем не так, как прогнозировалось в модели.

Кроме того, точность выходных данных модели не может быть лучше точности, с которой заданы входные параметры.

Модельный цикл отнюдь не является строго последовательным процессом. Между отдельными составляющими фаз валидации и проектирования могут существовать жесткие зависимости. Может потребоваться совместимость между описанием модели, замерами данных и методикой оценки модели. Достижение такой совместимости и ее согласование с конкретными целями моделирования являются по своей сущности процессами итерационными.[1]

**2.** **Описание системы**

В оптовом магазине используется новая процедура обслуживания клиентов. Клиенты, попадая в магазин, определяют по каталогу наименования товаров, которые они хотели бы приобрести. После этого клиента обслуживает клерк, который идет на расположенный рядом склад и приносит необходимый товар. Клиент ожидает дважды, сначала приема заказа, затем его выполнения. Каждый из клерков может обслуживать одновременно не более шести клиентов. Время, которое затрачивает клерк на путь к складу, равномерно распределено на интервале от 0.5 до 1,5 мин. Время поиска нужного товара зависит от числа наименовании, которые клерк должен найти на складе. Это время нормально распределено с математическим ожиданием, равным утроенному числу искомых наименований, и среднеквадратичным отклонением, равным одной пятой математического ожидания. Следовательно, если, например, со склада надо взять товар одного наименования, время на его поиск будет нормально распределено с математическим ожиданием, равным 3 мин. и среднеквадратичным отклонением, равным 36 с. Время возвращения со склада равномерно распределено на интервале от 0.5 до 1,5 мин. По возвращении со склада клерк рассчитывается со всеми клиентами, которых он обслуживает. Время расчета с клиентом равномерно распределено на интервале от 1 до 3 мин. Расчет производится в том порядке, в каком к клерку поступали заявки на товар. Интервалы между моментами поступления заявок на товары от клиентов экспоненциально распределены с математическим ожиданием, равным 2 мин Клиентов в магазине обслуживают три клерка. Цель моделирования — определить следующее:

О загрузку клерков;

О среднее время, необходимое на обслуживание одного клиента с момента подачи заявки на товар до оплаты счета за покупку;

О среднее число заявок, удовлетворяемых клерком за один выход на склад.

Продолжительность имитационного прогона составляет 1000 мин.

## 2.1 Модельное время

Так как время в задаче размерное, за единицу модельного времени примем секунд. Равномерное распределение будем генерировать непосредственно в секундах, а нормальное и экспоненциальное — в минутах, с последующим умножением на 60 и округлением до ближайшего целого.

## 2.2 Классы и объекты

В задаче описана открытая многоканальная система с неограниченным буфером, имеющая, однако, ряд довольно интересных особенностей. Обслуживание заявки в канале (клерком) представляет собой многоэтапный процесс с параметром - количеством единовременно обслуживаемых клиентов. Эта дисциплина носит название *групповое обслуживание*. Таким образом, текущее состояние процесса обслуживания характеризуется не одним значением — временем, оставшимся до завершения, а несколькими — номером этапа, временем, оставшимся до завершения этапа, и числом клиентов. Таких этапов четыре — путь на склад, поиск товара, путь обратно, расчет. На первых трех этапах число клиентов остается постоянным, на четвертом оно постепенно уменьшается до нуля, так как расплатившийся клиент покидает систему.

Интересна здесь также система очередей. Время пребывания клиента в магазине состоит из двух стадии. Сначала он стоит в общей очереди (назовем ее *первичной*) и ждет, когда один из клерков обратит на него внимание и примет заказ. Клиенты, находящиеся в первичной очереди, не связаны пока ни с каким клерком, а относятся как бы ко всему магазину в целом. После приема заказа клиент переходит в очередь, состоящую из людей, которые сделали заказ и ждут возвращения «своего» клерка со склада с товаром (назовем ее вторичной). Вторичная очередь соотносится с конкретным клерком, ее длина, согласно условию задачи, не может превышать шест, а количество вторичных очередей равно трем — общему количеству клерков. В противоположность этому первичная очередь может быть только одна и ограничений на длину не имеет. Разумеется, как первичная, так и любая из вторичных очередей может в течение некоторого времени быть пустой.

Заметим, что именно такая система обслуживания принята сейчас в большинстве магазинов, торгующих компьютерной и оргтехникой, в том числе и в том, услугами которого при необходимости пользуется автор. Некоторую аналогию можно провести и с обслуживанием в ресторане, гае клиент тоже сначала ждет прихода официанта, а затем — исполнения заказа, но понятие очереди в этом случае не столь акцентированное, да и система взаимоотношений официанта с клиентом ресторана все-таки несколько сложнее.

Все сказанное свидетельствует в пользу того, что логику работы клерка и всего магазина в целом надо отделить друг от друга и определить в разных классах, иначе сам принцип объектного моделирования будет выхолощен. Введем классы Клерк (Clerk) и Магазин (Shop)- Прежде чем перечислять их поля данных, подчеркнем следующее обстоятельство. В условии задачи дано максимальное значение объема группы - шесть. В общем случае можно ограничить и минимальное значение, которое назовем минимальным индексом группы (МИГ)- Смысл новою понятия заключается в то, что свободный клерк не начнет обслуживание клиентов до тех пор, пока длина первичной очереди не станет равна значению МИГ. Если к моменту накопления нужного количества клиентов свободных клерков несколько, выбор клерка, начинающего обслуживать эту группу, осуществляется случайным образом. Понятно, что стандартное значение МИГ — единица.

Еще один вопрос — взаимные ссылки между классами. В рассматриваемой системе объекты классов Clerk и Shop не являются равноправными, так как каждый из объектов класса Clerk входит в зону ответственности единственного объекта класса Shop, но не наоборот. Поскольку объект Shop управляет системой в целом, ему необходим доступ к любому объекту Clerk для передачи ему различных сообщений (например, указание принять заказ). Каждый из клерков отвечает только за себя, и ему ссылка на Shop не нужна, так как всем информационным обменом руководит Shop. Поскольку перекрестных ссылок пет, тип указателя при объявлении поля класса Shop можно указывал, в явном виде (Clerk\*\*), если, конечно, класс Clerk описан в header-файле раньше, чем класс Shop.

Перечислим поля данных класса Clerk.

Неизменяемые поля:

* среднее время нахождения клерка в пути (60 с);
* максимальное отклонение от среднего для времени нахождения клерка в пути (30 с);
* среднее время расчета одного клиента (120 с);
* максимальное отклонение от среднего для времени расчета одного клиента (60 с);
* уникальный номер клерка.

Изменяемые поля:

* вторичная очередь. Моделируется массивом указателей на объекты класса Client. Если клерк свободен, очередь пуста;
* клиент, с которым в данный момент производится расчет. Поле данных имеет смысл только при нахождении клерка в состоянии Расчет;
* текущее число клиентов, у которых принят заказ и которые ожидают возвращения клерка. Не то же самое, что длина вторичной очереди, поскольку в процессе расчета клиентов длина очереди меняется. Эго поле данных характеризует именно размер пакета заказов. Равен -1, если клерк свободен;
* время, оставшееся до прибытия клерка на склад. Значение поля данных активно только в состоянии движения на склад за товаром. В любом другом состоянии равно -1;
* время, оставшееся до возвращения клерка со склада Значение поля данных активно только в состоянии движения со склада с товаром. В любом другом состоянии равно -1;
* время, оставшееся до окончания расчета текущего клиента. Значение поля данных активно только в состоянии расчета клиентов. В любом другом состоянии рано -1;
* время, оставшееся до завершения поиска товаров. Значение поля данных активно только в состоянии поиска товаров при нахождении клерка на складе. В любом другом состоянии равно -1;
* время, прошедшее с момента принятия заказа Поле данных необходимо для сбора статистики о длительности цикла клерка — от принятия заказа до расчета последнего клиента. Если клерк свободен, значение равно -1.

Поля данных класса Shop. Неизменяемые поля:

* количество клерков (3). Для удобства реализации сделано глобальной переменной;
* максимальный объём одного заказа (6). Для удобства реализации сделано глобальной переменной;
* минимальный индекс группы (1);
* средняя интенсивность входного потока (0,5 заявок в минуту);
* массив указателей на объекты класса Clerk.

Изменяемые поля:

* первичная очередь клиентов. Из-за отсутствия ограничений на максимальную длину моделируется связным списком;
* время, оставшееся до прибытия следующей заявки из входного потока;
* текущая длина первичной очереди (вычисляемое поле).

Отношения дружественности между классами построены следующим образом: друзьями класса Client являются Clerk и Shop, другом класса Clerk - класс Shop.

## 2.3 События и методы

Каждому из пяти возможных состояний клерка соответствует событие, в результате которого он покидает это состояние и переходит в другое. Каждому событию, в свою очередь, сопоставлен отдельный метод. Перечислим эти события:

1. Прибытие клерка на склад.
2. Завершение поиска заказанного товара.
3. Прибытие клерка с товаром к ожидающим ею клиентам.
4. Завершение расчетов с очередным клиентом.
5. Принятие заказа у клиентов из первичной очереди.

Подробнее остановимся на реализации последнего метода. Если первичная очередь не пуста и ее длина достигла значения МИГ, объект Shop пытается препоручить как можно больше клиентов одному из свободных клерков. После того как клерк выбран, ему посылается сообщение, соответствующее методу 5, с двумя параметрами: указателем на первичную очередь, чтобы клерк мог скопировать часть ее клиентов во вторичную, и количеством клиентов, заказы у которых магазин предписывает принять клерку. Возвращает же он объекту Shop указатель на клиента первичной очереди, который теперь становится в этой очереди первым, то есть на новую голову связного списка. Первичную очередь Shop продвигает сам. Все эти действия выполняет метод-диспетчер run().

Методы класса Shop:

* прибытие нового клиента из внешнего потока и постановка ею в первичную очередь;
* выбор клерка, который должен принять заказ. Метод выбирает случайным образом одного клерка из числа свободных в данный момент.

**3. Реализация модели**

## 3.1 Программная реализация

Имитационное моделирование это процесс конструирования модели реальной системы и постановки экспериментов на этой модели с целью либо понять поведение системы, либо оценить (в рамках ограничений) различные стратегии, обеспечивающие функционирование системы. Имитационное моделирование является экспериментальной и прикладной методологией, которая:

* описывает поведение системы;
* строит теории и гипотезы, которые могут объяснить наблюдаемое поведение;
* использует эти теории для предсказания будущего поведения системы, то есть тех воздействий, которые могут быть вызваны изменениями в системе или изменениями способов ее функционирования.[1]

При выполнении данной программы генерируются число занятых клерков, среднее время периода занятости клерка, средняя длина первичной очереди, среднее число клиентов в магазине, среднее время пребывания клиента в магазине, средний объем одного заказа. Другими словами, в имитационном эксперименте входные данные «пропускаются» через логическую структуру, чей ответ «подражает» ответы реальной системы на входные данные. Требуется составить алгоритм, и реализовать его. Для моделирования системы группового обслуживания с несколькими этапами и двойной очередью (работа оптового магазина), был выбран высокоуровневый язык программирования C++ и написана программа на этом языке, позволяющая в полной мере отразить функционирование системы. Для проведения анализа зависимостей некоторых показателей друг от друга была использована программа gnuplot.

**3.2 Построение графиков**

### 3.2.1 Программа gnuplot

Gnuplot портативная программа для визуализации данных и создания графиков функций для операционных систем UNIX, IBM OS/2, MS Windows, DOS, Macintosh, VMS, Atari и многих других. Эта программа защищена авторским правом, но свободна для распространения.

Gnuplot поддерживает множество видов графиков как двух-, так и трехмерных. Он может рисовать, используя линии, точки, боксы, контуры, векторные поля, поверхности и различный связанный текст.[2]

Gnuplot имеет собственную систему команд, может работать интерактивно (в режиме командной строки) и выполнять скрипты, читаемые из файлов. Также используется в качестве системы вывода изображений в различных математических пакетах: GNU Octave, Maxima и других. [3]

Gnuplot поддерживает много различных форматов для выдачи: интерактивные графические терминалы (с мышью и функциями горячих клавиш), прямой вывод на плоттеры и современные принтеры, запись в различные форматы файлов (eps, fig, jpeg, LaTeX, metafont, pbm, pdf, png, postscript, svg и так далее). Gnuplot легко расширяем для включения новых функций. [2]

### 3.2.2 Использование программы для построения графиков

Для того, чтобы нарисовать график, достаточно указать набор команд в тэгах <plot>...</plot>. Основные команды состоят из задания области определения функции (для одномерных графиков это переменная «x», для двухмерных «x», «y»), и команды отрисовки одномерной или двухмерной функции, заданной в символьном виде. Синтаксис функции интуитивно понятен, «+», «-», «\*», «/» обозначают стандартные арифметические операторы (умножение должно быть явным, никаких математических сокращений типа «3x» и т. п.), «\*\*» означает возведение в степень, скобки "(", «)» используются для задания приоритета.

Кроме операторов, есть набор стандартных математических функций:

Тригонометрические функции sin, cos, tan, константа pi, и им обратные asin, acos, atan.

Гиперболические функции sinh, cosh, tanh.

Экспонента exp и натуральный и десятичный логарифмы: log и log10 соответственно.

Трехмерные графики рисуются аналогично, нужно задать диапазоны для области определения и использовать команду «splot». [4]

**4.** **Анализ результатов**

Цифровые данные, полученные при 1000-минутном моделировании. Усредненные результаты:

* Количество поступлений – 477;
* Обслужено клиентов – 460;
* Средне число занятых клерков – 2,65
* Средняя длительность периода занятости клерка – 11,84 мин;
* Средняя длина первичной очереди – 1,31;
* Среднее число клиентов в магазине – 8,55;
* Среднее время пребывания клиента в магазине – 18,04 мин;
* Средний объем одного заказа – 2,026;
* Среднее время пребывания в первичной очереди – 18,04-11,84=6,2 мин.

На рис. 1-3 приведены графики зависимости от числа клерков, соответственно, загрузки системы, среднего числа клерков и среднего времени пребывания клиента в системе. Из графиков видно, что оптимальное число клерков – четыре. Дальнейшее увеличение числа клерков к значимому улучшению показателей функционирования не приводит. Добавление же четвертого клерка все еще позволяет существенно улучшить эти показатели. Приведем их:

* Средне число занятых клерков – 3,32
* Средняя длительность периода занятости клерка – 11,78 мин;
* Средняя длина первичной очереди – 1,192;
* Среднее число клиентов в магазине – 8,451;
* Среднее время пребывания клиента в магазине – 16,05 мин;
* Средний объем одного заказа – 1,983;
* Среднее время пребывания в первичной очереди –

16,05-11,78=4,27 мин.

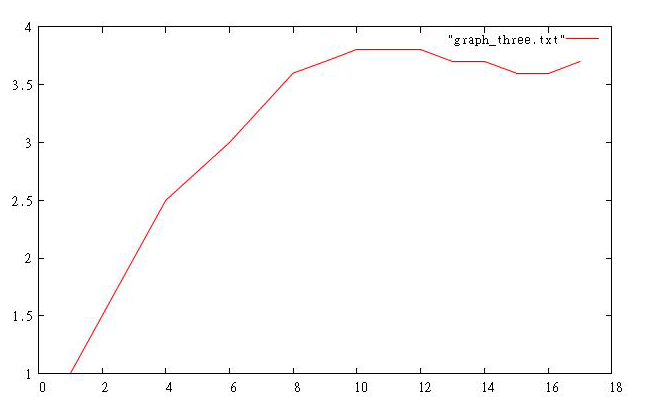


Рис. 1 Зависимость коэффициентов загрузки от числа клерков

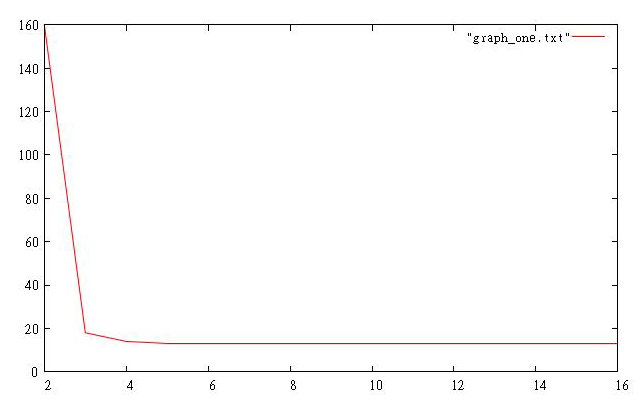


Рис. 2 Зависимость среднего числа клиентов в магазине от числа клерков

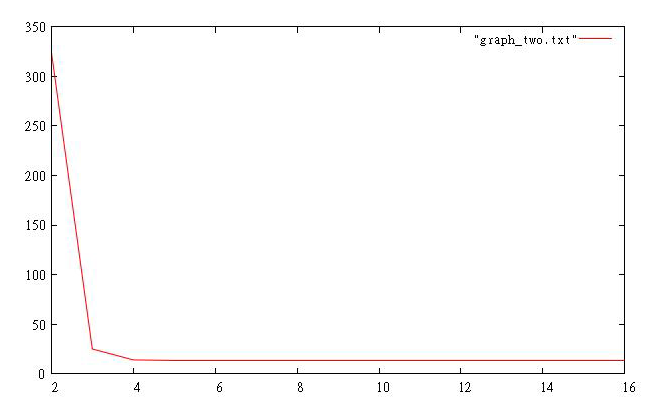


Рис. 3 Зависимость среднего времени пребывания клиента в магазине от числа клерков

**Заключение**

В результате выполнения курсовой работы были достигнуты следующие результаты:

* изучены методы построения имитационных моделей экономических объектов;
* получены навыки проведения численных экспериментов на имитационных моделях экономических систем;
* приобретен опыт проведения анализа по результатам численных экспериментов на имитационной модели;
* проведенный анализ позволил обнаружить некоторые закономерности, которые помогут в проведении кадровой политике предприятия.

**Список использованной литературы**

1. Труб И. И. «Объектно-ориентированное моделирование на С++»: Учебный курс.-СПб.:Питер, 2006.-411с.:ил.
2. Gnuplot. Описание программы. Режим доступа: http://fsweb.info/calculations/gnuplot.html
3. Построение графиков в gnuplot. Режим доступа:

http://ru.wikipedia.org/wiki/Gnuplot

1. Gnuplot. Режим доступа: http://lib.custis.ru/Gnuplot