

Содержание:

Введение

В данной работе необходимо промоделировать работу бухгалтерии предприятия. Постановка задачи звучит следующим образом.

В бухгалтерии предприятия работает один бухгалтер, который имеет персональный компьютер с установленным пакетом программ. С интервалом, распределенным по нормальному закону со средним значением 6 мин и среднеквадратичным отклонением 2,0 мин, курьер приносит в бухгалтерию документы, которые бухгалтер обрабатывает на компьютере в порядке их поступления. Время обработки документа бухгалтером распределено по нормальному закону со средним значением 4,8 мин и среднеквадратичным отклонением 1,6 мин. Обработанные документы подшиваются в специальную папку. В работе компьютера периодически возникают неполадки (неисправность в аппаратной части, сбой в программе, нарушение питания и т.п.), в случае чего обработка документов прекращается и вызывается специалист, который устраняет неисправность. Интервал времени между возникновениями неполадок распределен экспоненциальному закону со средним 24 ч, устранение неполадки требует $1 \pm 0,5$ ч. Считается, что устранение неполадки начинается мгновенно после ее возникновения. После устранения неполадка учитывается в специальном журнале. Курьер продолжает приносить документы и во время ремонта компьютера.

Необходимо смоделировать работу бухгалтерии в течение 1200 часов и определить:

- среднее время ожидания документами обслуживания.
- среднюю величину очереди на обслуживание,
- среднюю загрузку бухгалтера,

после чего получить значения тех же параметров на модели безотказного оборудования и сравнить результаты.

Задача будет решена с помощью имитационного моделирования, которое обладает следующими преимуществами:

Применение имитационных моделей дает множество преимуществ по сравнению с выполнением экспериментов над реальной системой и использованием других методов.

- **Стоимость.** Допустим, компания уволила часть сотрудников, что в дальнейшем привело к снижению качества обслуживания и потери части клиентов. Принять обоснованное решение помогла бы имитационная модель, затраты на применение которой состоят лишь из цены программного обеспечения и стоимости консалтинговых услуг.
- **Время.** В реальности оценить эффективность, например, новой сети распространения продукции или измененной структуры склада можно лишь через месяцы или даже годы. Имитационная модель позволяет определить оптимальность таких изменений за считанные минуты, необходимые для проведения эксперимента.
- **Повторяемость.** Современная жизнь требует от организаций быстрой реакции на изменение ситуации на рынке. Например, прогноз объемов спроса продукции должен быть составлен в срок, и его изменения критичны. С помощью имитационной модели можно провести неограниченное количество экспериментов с разными параметрами, чтобы определить наилучший вариант.
- **Точность.** Традиционные расчетные математические методы требуют применения высокой степени абстракции и не учитывают важные детали. Имитационное моделирование позволяет описать структуру системы и её процессы в естественном виде, не прибегая к использованию формул и строгих математических зависимостей.
- **Наглядность.** Имитационная модель обладает возможностями визуализации процесса работы системы во времени, схематичного задания её структуры и выдачи результатов в графическом виде. Это позволяет наглядно представить полученное решение и донести заложенные в него идеи до клиента и коллег.
- **Универсальность.** Имитационное моделирование позволяет решать задачи из любых областей: производства, логистики, финансов, здравоохранения и многих других. В каждом случае модель имитирует, воспроизводит, реальную жизнь и позволяет проводить широкий набор экспериментов без влияния на реальные объекты.

Подходы к решению задачи

Основными методами имитационного моделирования являются: аналитический метод, метод статического моделирования и комбинированный метод (аналитико-статистический) метод.

Аналитический метод применяется для имитации процессов в основном для малых и простых систем, где отсутствует фактор случайности. Например, когда процесс их функционирования описан дифференциальными или интегро-дифференциальными уравнениями. Метод назван условно, так как он объединяет возможности имитации процесса, модель которого получена в виде аналитически замкнутого решения, или решения полученного методами вычислительной математики.

Метод статического моделирования первоначально развивался как метод статистических испытаний (Монте-Карло). Это - численный метод, состоящий в получении оценок вероятностных характеристик, совпадающих с решением аналитических задач (например, с решением уравнений и вычислением определенного интеграла). В последствии этот метод стал применяться для имитации процессов, происходящих в системах, внутри которых есть источник случайности или которые подвержены случайным воздействиям. Он получил название метода статического моделирования.

Комбинированный метод (аналитико-статистический) позволяет объединить достоинства аналитического и статического методов моделирования. Он применяется в случае разработки модели, состоящей из различных модулей, представляющих набор как статистических, так и аналитических моделей, которые взаимодействуют как единое целое. Причем в набор модулей могут входить не только модули соответствующие динамическим моделям, но и модули соответствующие статическим математическим моделям.

В математических моделях сложных объектов, представленных в виде систем массового обслуживания (СМО), фигурируют средства обслуживания, называемые обслуживающими аппаратами (ОА) или каналами, и обслуживаемые заявки, называемые транзактами.

Обоснование выбора системы имитационного моделирования

В качестве типичных представителей поколения систем имитационного моделирования 90-х гг. можно назвать следующие распространенные пакеты:

- 1) Process Charter-1.0.2 (генеральный разработчик и поставщик системы - компания "Scitor", г. Менло-Парк, шт. Калифорния, США);
- 2) Powersim-2.01 (генеральный разработчик и поставщик вариантов системы - компания "Modell Data" AS, г. Берген, Норвегия);
- 3) Lthink-3.0.61 (генеральный разработчик и поставщик системы - компания "High Performance Systems", г. Ганновер, шт. Нью-Хэмпшир, США);
- 4) Extend+BPR-3.1 (генеральный разработчик и поставщик различных вариантов системы - компания "Imagine That!", г. Сан-Хосе, шт. Калифорния, США).
- 5) ReThink (генеральный разработчик и поставщик англоязычного варианта системы - фирма "Gensym", г. Кембридж, шт. Массачусеттс, США);
- 6) Pilgrim-2.1 (генеральные соразработчики и поставщики системы - фирма "МегаТрон", г. М., Россия; компания "Keisy", г. Гаага, Нидерланды, компания "Enit" AS, г. Таллинн, Эстония);

Пакет Process Charter-1.0.2 имеет "интеллектуальное" средство построения блок-схем моделей. Он ориентирован в основном на дискретное моделирование. Имеет достоинства: удобный и простой в использовании механизм построения модели, самый дешевый из перечисленных продуктов, хорошо приспособлен для решения задач распределения ресурсов. Недостатки пакета: наименее мощный продукт, слабая поддержка моделирования непрерывных компонентов, недостаток средств для анализа чувствительности и построения диаграмм. Пакет Powersim-2.01 является хорошим средством создания непрерывных моделей. Имеет достоинства: множество встроенных функций, облегчающих построение моделей, многопользовательский режим для коллективной работы с моделью, средства обработки массивов для упрощения создания моделей со сходными компонентами. Недостатки пакета: сложная специальная система обозначений System Dynamics, ограниченная поддержка дискретного моделирования. Пакет Lthink-3.0.61 обеспечивает создание непрерывных и дискретных моделей. Имеет достоинства: встроенные блоки для облегчения создания различных видов моделей, поддержка авторского моделирования слабо подготовленными пользователями, подробная обучающая программа, развитые средства анализа чувствительности, поддержка множества форматов входных данных. Недостатки пакета: сложная система

обозначений System Dynamics, поддержка малого числа функций по сравнению с Powersim-2.01. Пакет Extend+BPR-3.1 создан как средство анализа бизнес-процессов (сокращение BPR означает Business Process Reengineering), широко использовался NASA, поддерживает дискретное и непрерывное моделирование. Имеет достоинства: интуитивно понятная среда построения моделей с помощью блоков, множество встроенных блоков и функций для облегчения создания моделей, поддержка сторонними компаниями (особенно выпускающими приложения для “вертикальных” рынков), гибкие средства анализа чувствительности, средства создания дополнительных функций с помощью встроенного языка. Недостатки пакета: используется в полном объеме только на компьютерах типа Macintosh, имеет высокую стоимость. Пакет ReThink обладает свойствами Extend+BPR-3.1 и в отличие от перечисленных пакетов имеет хороший графический транслятор для создания моделей. Выполняется под управлением экспертной оболочки G2. Имеет достоинства: все положительные свойства Extend+BPR-3.1 и общее поле данных с экспертной системой реального времени, создаваемой средствами G2. Недостатки пакета: требует применения станций большой мощности, требует экспертную оболочку G2, которая далеко не всегда необходима (особенно при создании моделей бизнес-процессов в ускоренном масштабе времени), имеет слабую поддержку создания непрерывных компонент модели и высокую стоимость.

Попытки создания универсальной системы имитационного моделирования привели к появлению системы Pilgrim-2.1, которая стала распространяться в России, Нидерландах и странах Балтии в начале 90-х гг. Пакет Pilgrim обладает широким спектром возможностей имитации временной, пространственной и финансовой динамики моделируемых объектов. С его помощью можно создавать дискретно-непрерывные модели. Разрабатываемые модели имеют свойство коллективного управления процессом моделирования. В текст модели можно вставлять любые блоки с помощью стандартного языка C. Различные версии этой системы работают на DEC-совместимых и IBM-совместимых компьютерах, оснащенных Unix, MS DOS (и Windows). Pilgrim обладает свойством мобильности, т.е. переноса на любую другую платформу при наличии ОС Unix и компилятора C++. Модели в системе Pilgrim компилируются и поэтому имеют высокое быстродействие, что очень важно для отработки управленческих решений и адаптивного выбора вариантов в сверхускоренном масштабе времени. Полученный после компиляции объектный код можно встраивать в разрабатываемые программные комплексы, так как при эксплуатации моделей инструментальные средства пакета Pilgrim не используются. Стоимость пакета существенно ниже суммарной стоимости ReThink и G2.

Недостаток пакета: отсутствие такого мощного графического транслятора, какой есть в пакете ReThink (однако такой графический транслятор не всегда бывает необходим).

В качестве системы имитационного моделирования была выбрана система Pilgrim, в связи с такими его достоинствами, как возможность имитации финансовой динамики объектов и быстрое действие.

Имитационная модель

Граф модели Pilgrim

Средством построения моделей в системах типа PILGRIM является специфический конструктор, представляющий собой набор узлов различного назначения. На схеме имитационной модели узел есть графическое изображение некоторого типового процесса, или можно сказать, что внутри узла работает процесс. При этом элементарный процесс может быть, вообще говоря, представлен несколькими узлами имитационной модели, если этого требует логика его работы.

Таким образом, схема имитационной модели представляет собой направленный граф, вершины которого представляют собой компоненты элементарных процессов, а дуги определяют направление потоков заявок.

В системе имитационного моделирования PILGRIM граф модели описывает взаимодействия дискретных и непрерывных компонент имитационной модели и представляет собой стохастическую сеть. Каждый узел графа – это ветвь моделирующей программы.

Непрерывные компоненты представляют собой разностные уравнения, причем каждый очередной интервал (или шаг) интегрирования – это отрезок времени между двумя ближайшими событиями в стохастической сети.

Одновременность моделирования всех компонент системы имитируется с помощью единой службы модельного времени.

Основной динамической единицей любой модели, работающей под управлением имитатора, является транзакт.

Транзакт – это формальный запрос на какое-либо обслуживание, например сигнал о загрязнении какого-то пункта местности, телеграмма, поступающая на узел коммутации сообщений, проба загрязненной почвы, ожидающая соответствующего анализа и т.д. Пути «миграции» транзактов по графу определяются логикой функционирования дискретных компонентов модели.

Узлы графа представляют собой центры обслуживания транзактов. В узлах транзакты могут задерживаться, обслуживаться, порождать семейства новых транзактов, уничтожать другие транзакты. Вид обслуживания транзакта определяется типом узла.

Например, предприятие, на котором происходят аварийные выбросы загрязняющих веществ, - это узел, в котором один транзакт порождает случайное число других транзактов, рассеиваемых по координатной сетке местности в случайные моменты модельного времени. Другой пример узла - это передвижная лаборатория, в которой есть один транзакт, представляющий собой распоряжение об обследовании местности, и этот транзакт должен собирать другие - сигналы о возможных недопустимых загрязнениях природной среды. Третий пример - телеграмма, поступившая в центр коммутации сообщений и ожидающая в памяти связанной ЭВМ дальнейшей отправки; в данном примере узел - это связанная ЭВМ.

Нумерация узлов графа производится разработчиком модели. Следует учесть, что транзакт всегда принадлежит одному из узлов графа и, независимо от этого, относится к определенной точке пространства или местности, координаты которой могут изменяться.

Выход из узла одного транзакта называется событием. Предположим, что в момент времени t произошло какое-то событие, а в момент времени $t+d$ должно произойти ближайшее следующее, но не обязательно в этом же узле. Тогда очевидно, что передать управление непрерывным компонентам модели можно только на время $[t, t+d]$.

Текущее значение модельного таймера хранится в системной переменной `timer`. Значение модельного таймера доступно для чтения в процессе моделирования.

Вид обслуживания транзакта определяется типом узла.

Существуют следующие типы узлов:

- Узел AG - постоянный генератор транзактов

- Узел SERV - обслуживающий многоканальный прибор с абсолютными приоритетами.
- Узел QUEUE - очередь с относительными приоритетами.
- Узел TERM - безусловный терминатор транзактов.
- Узел KEY - клапан на пути транзактов.
- Узел DELET - управляемый терминатор транзактов (демультипликатор).
- Узлы SEND и DIRECT - операции с бухгалтерскими счетами.

Для данной задачи получен следующий граф (рис.1).

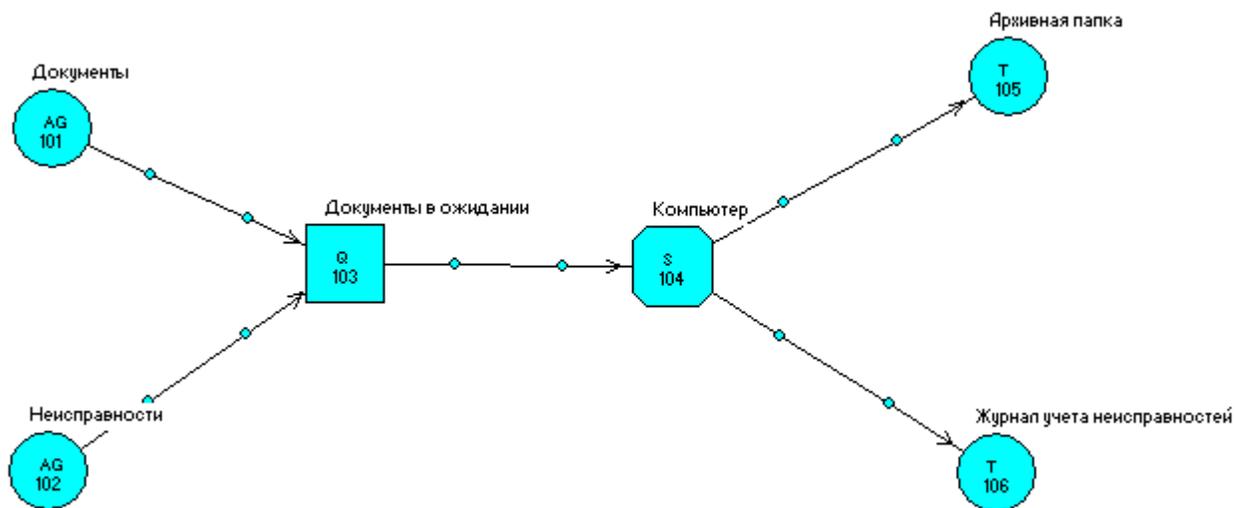


Рис.1. Граф модели

Описание модели

В модели имеются транзакты двух типов:

- бухгалтерские документы;
- неполадки в работе компьютера.

Транзакты-документы создаются генератором 5. Они попадают в очередь 1 (имитация документов, лежащих на столе в ожидании обработки) и далее в сервер 2, имитирующий компьютер бухгалтера. Из сервера транзакты - документы переходят в терминатор 3, имитирующий папку, куда подшиваются документы.

Транзакты-неполадки создаются генератором 6. Они также проходят по узлам 1 и 2 и попадают в терминатор 4, имитирующий журнал учета неполадок.

Обслуживание транзакта-документа состоит во вводе информации в компьютер и требует времени “ t ввода”. Обслуживание транзакта-неполадки состоит в ее устранении и требует времени “ t ремонта”. Таким образом, сервер, моделирующий компьютер, будет менять время обслуживания транзакта в зависимости от типа пришедшего транзакта.

Различать транзакты будем по их приоритетам, которые зададим при описании соответствующих генераторов. Транзакты-неполадки будут иметь более высокий приоритет - 1, а транзакты-документы - 0. Приоритет будет использоваться также для того, чтобы заблокировать обработку документов на время ремонта компьютера. Для этого зададим в параметрах сервера и очереди к нему флажки учитывания приоритетов транзактов (значения “abs” и “prty”). Тогда очередь будет ставить транзакты-неполадки впереди транзактов-документов, а сервер будет работать в режиме прерывания обслуживания менее приоритетных транзактов.

Если в сервере находится транзакт-документ (то есть происходит ввод информации), и в это время появляется транзакт-неполадка, она автоматически становится первой в очереди. Сервер “видит” в начале очереди более приоритетный транзакт и вытесняет транзакт-документ в специальный стек (ввод информации прекращается по причине сбоя в работе компьютера). Транзакт-неполадка занимает сервер (начинается процесс устранения неполадки) и находится там до истечения заданного времени ремонта. После этого продолжается обслуживание вытесненного транзакта-документа (ввод оставшейся информации по прерванному документу).

Если мы зададим учет приоритетов транзактов только для очереди, то сервер не будет работать в режиме прерывания обслуживания транзактов, что соответствует условию “компьютер никогда не ломается во время обработки документа”. Такая конфигурация узлов подходит, например, для моделирования работы продавца, который обслуживает некоторую категорию покупателей вне очереди, но не бросает обслуживание текущего покупателя при появлении “более приоритетного”.

Разберем выбор законов распределения случайных параметров модели - интервалов работы генераторов и времен обслуживания сервера. В таблице заданы средние значения этих параметров, а реальные значения для каждого нового

транзакта будут иметь некоторый разброс около этих средних.

Интервал работы генератора транзактов-документов имеет нормальный закон распределения (norm). Мы исходим из того, что случайные значения промежутка времени между приходами курьера группируются около известного среднего. Если бы в условии было сказано, что бухгалтерию обслуживают несколько курьеров, то каждый из них имел бы свой собственный график посещений бухгалтерии со случайными (совершенно неизвестными нам) промежутками между приходами. В этом случае, на основании теоремы о суперпозиции случайных потоков событий (в нашем случае - приходов курьера), интервал генератора транзактов-документов имел бы экспоненциальный закон распределения (expo). Время обслуживания сервером транзакта-документа (время ввода информации в компьютер) имеет нормальный закон распределения. Мы исходим из того, что поступающие на обработку документы имеют один и тот же тип, и отклонение во времени ввода информации связано только с тем, что бухгалтер-оператор не может работать так же точно, как автомат.

Интервал генератора транзактов - неполадок имеет экспоненциальный закон распределения. Компьютер имеет очень большое число деталей, каждая из которых имеет свой срок наработки на отказ (интервал между выходами из строя). Время обслуживания транзакта-неполадки распределено по равномерному закону (unif) в пределах от 0.5 часа до 1.5 часа. Мы считаем, что устранение неисправности может с равной вероятностью занять любой промежуток времени от получаса до полутора часов.

Перед тем, как очередной транзакт войдет в сервер, должны быть соответствующим образом настроены параметры сервера. Это делается с помощью условного оператора "if" и операторов присваивания. В описании сервера его меняющие значение параметры заданы с помощью переменных пользователя. Вид закона распределения времени обслуживания (Dist) и номер терминатора, в который должен перейти транзакт после выхода из сервера (Forw) имеют целый тип (int). Математическое ожидание (среднее) времени обслуживания (Tobs) и среднеквадратическое отклонение (Pogr) являются переменными с плавающей точкой (float). Для равномерного закона вместо математического ожидания задается середина допустимого интервала значений, а вместо отклонения - ширина этого интервала. Для нормального закона среднеквадратическое отклонение можно всегда задавать как одну треть от математического ожидания. Для экспоненциального закона отклонение всегда равно математическому ожиданию, поэтому в моделях можно писать вместо него константу "zero", а ПИЛИГРИМ сам подставит нужное значение.

Условие, проверяемое при настройке параметров, есть равенство приоритета транзакта ($t \rightarrow pr$) единице, что соответствует приходу транзакта-неполадки. По первой ветке параметры сервера настраиваются на обработку транзакта-неполадки, а по второй - документа.

Исходный текст программной модели

```
#include <pilgrim.h>

int Forw; /* Номер следующего узла */

int Dist; /* Закон распределения времени обслуживания */

float Tobs; /* Среднее время обслуживания */

float Pogr; /* Среднеквадратическое отклонение */

/* Время измеряется в часах */

forward

{

modbeg("Компьютер", 6, 1200.0, (long)time(NULL), none, 1, none, 3, 2);

ag("Бухгалтерия", 5, none, expro, 0.1, 0.1/3, zero, 1);

ag("Неисправности", 6, 1, expro, 24.0, zero, zero, 1);

network(dummy, dummy)

{

top(1): queue("Столик-очередь", prty, 2);

place;

top(2): if (t->pr == 1) /* Неполадка */

{

Forw=4;

Dist=unif;
```

```
Tobs=1.0;

Pogr=0.5;

}

else /* Документ */

{

Forw=3;

Dist=norm;

Tobs=0.08;

Pogr=0.08/3;

}

serv("АРМ бухгалтерии", 1, abs, norm, Tobs, Pogr, zero, Forw);

place;

top(3): term("Архивные папки");

place;

top(4): term("Неиспр. устр-на");

place;

fault(123);

}

modend("pilgrim.rep", 1, 12, page);

return 0;

}
```

Анализ полученных результатов

На рисунке 2 приведены результаты моделирования.

НАЗВАНИЕ МОДЕЛИ: АРМ бухгалтера		Лист: 1									
ВРЕМЯ МОДЕЛИРОВАНИЯ: 1200.07											
No узла	Наименование узла	Тип узла	Точка ка	Загрузка, %	М [т]	С [т]	Счетчик входов	Кол. зам.	Ост. тр.	Состояние узла в этот момент	
1	Столлик-очередь	queue	-	-	0.37	2.02	12094	1	1	открыт	
2	АРМ бухгалтера	serv	-	87.4	0.09	1.90	12093	1	0	открыт	
3	Арифметические пакеты	term	-	-	0.45	1.38	12044	0	0	открыт	
4	Неиспр. устр-на техн	term	-	-	1.00	0.16	49	0	0	открыт	
5	Бухгалтерия	ag	-	-	0.10	0.24	12045	1	1	открыт	
6	Неисправности	ag	-	-	23.68	1.05	50	1	1	открыт	

Рис.2. Результаты моделирования

В таблице результатов прогона модели мы видим статистические данные о работе бухгалтера за период, равный 1200 часов. За это время средняя загрузка бухгалтера составила почти 90 % (загрузка узла 2), что довольно много. Среднее время ожидания документом обработки составило примерно 0.4 часа (M [t] узла 1), что также довольно много. Это происходит потому, что интервал поступления документов почти совпадает с временем их обработки. При этом из-за отклонений в значениях времен образуются очереди. По значению квадрата коэффициента вариации (вариация узла 1) можно сказать, что это время имело разброс значений, чуть больший, чем матожидание (корень из к-та вариации чуть больше 1).

При анализе состояния очередей особое внимание следует обратить на значение **квадрата коэффициента вариации** времени задержки. Если оно близко к нулю, время задержки не имело существенного разброса, то есть длина очереди была практически постоянной. Если же значение квадрата коэффициента вариации велико, это значит, что транзакты приходили в очередь **группами**. Средний размер группы равен корню из квадрата коэффициента вариации. В этом случае нужно установить причину образования групп транзактов на входе очереди. Это может быть, например, ключ или многоканальный сервер, стоящий перед очередью.

Всего было обработано 12044 документа (счетчик входов узла 3) и устранено 49 неполадок (счетчик входов узла 4). Один документ остался необработанным (остаток транзактов узла 1). Все возникшие неполадки устранены. Оставшийся в очереди транзакт есть именно документ, а не неполадка, так как всего было сгенерировано 49 транзактов-неполадок (последний, оставшийся в генераторе, не

считается), и все они добрались до своего терминатора.

Заключение

Одной из наиболее крупных отраслей развития технологий с применением ЭВМ, является математическое моделирование, которое достаточно просто (в отличие от аналогового моделирования) может быть реализовано на ЭВМ разных модификаций и возможностей. Математическое моделирование тесно связано с имитационным моделированием. Одним из разделов математического моделирования являются модели систем массового обслуживания и их изучение.

В данном курсовом проекте была построена имитационная модель работы бухгалтерии предприятия с использованием программы Pilgrim. Результатом данного курсового проекта является имитационная модель работы бухгалтерии. Модель реализована на языке программирования высокого уровня Си++.

Программа является рабочей. Выдает на экран вероятностные и статистические характеристики работы процесса обслуживания посетителей предприятия.

На основании проделанной работы, можно сделать следующие выводы:

1. Математическая модель системы массового обслуживания, созданная нами, адекватна реальному объекту;
2. Проведенные исследования показали эффективность нашей модели и способов «приведения её в действие» при определении необходимых нам параметров по сравнению с ручным способом моделирования и расчетов параметров;
3. Созданная модель имеет достаточную, для таких моделей, степень универсальности, т.к. диапазон входных параметров системы можно легко и быстро изменить.

Список использованной литературы

1. Варфоломеев В.И. «Алгоритмическое моделирование элементов экономических систем». - М.: Финансы и статистика, 2000г.
2. Емельянов А.А., Власова Е.А. Имитационное моделирование экономических процессов. - М. Московский международный институт эконометрики, информатики, финансов и права. 2002. – 92 с.

3. Кеольтон В., Лод А. «Имитационное моделирование. Классика CS» издание 3-е, 2004г.;
4. Прицкер А. «Введение в имитационное моделирование»-М.: Мир,1987.-644с.
5. Шеннон Р. Имитационное моделирование систем: искусство и наука.«Мир», М.:1978.418 с.