

Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

И. А. Веренинов

"Алгоритмизация и основы программирования"

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Санкт-Петербург
2016

Аннотация.

Методические указания предназначены для выполнения студентами первого курса курсового проекта по дисциплине "Алгоритмизация и основы программирования". Эта дисциплина изучается в соответствии с программой обучения бакалавров по направлению "Программная инженерия". Методические указания обеспечивают качественное выполнение первого крупного самостоятельного творческого проекта поскольку содержат не только важные принципы моделирования дискретных стохастических систем, но и конкретные требования по процедуре проектирования подобных систем, включая подробное описание выполненного программного продукта с интерпретацией полученных результатов.

Содержание:

1. Введение;
2. Пример задания.
3. Этапы разработки алгоритма моделирования;
4. Определение функций блоков БМС;
5. Детализация блоков БМС;
6. Рекомендации по организации вывода промежуточных данных в алгоритме;
7. Определение длины реализации;
8. Состав программного продукта курсового проекта.
9. Рекомендации по разработке модулей;
10. Требования к исходным текстам и файлу Readme.doc.
11. Порядок разработки.
12. Оценка точности определения характеристик моделируемой системы методом статистического моделирования.

ВВЕДЕНИЕ.

Основная цель курсовой работы - получение практических навыков объектного программирования при решении задач моделирования систем, формализованных как системы массового обслуживания. При этом не используются какие-либо специализированные системы моделирования, а алгоритм программируется на языке Pascal и реализуется на ПК. Относительно большая степень детализации описания процесса функционирования системы в этом случае позволяет достичь и дополнительной цели, а именно, на достаточно сложных алгоритмах закрепить приобретённые студентами навыки разработки и отладки на ПК многомодульных программ, используя ручную объектно-ориентированную технологию.

Объект исследования формализован как СМО с многомерным входным потоком, памятью заявок, заданной дисциплиной записи заявок в память и выборки их на обслуживание, одним прибором и одной фазой обслуживания.

Цель разрабатываемого программного продукта - исследование таких систем для оценки стационарных характеристик их эффективности в зависимости от параметров элементов систем при заданной их структуре .

Исследуемые студентами системы одинаковы по сложности и отличаются:

1. Числом источников заявок, функциями распределения интервалов между заявками, приоритетностью источников заявок;
2. Дисциплиной записи заявок в память (буфер), объёмом буфера;
3. Дисциплиной выборки заявок из буфера на обслуживающий прибор;
4. Законами распределения длительности обслуживания.

К варьируемым в задании параметрам элементов системы относятся параметры законов распределения интервалов между заявками и закона обслуживания заявок прибором.

В качестве характеристик эффективности функционирования используются вероятность отказов от обслуживания заявок по каждому из источников, среднее время ожидания заявок в очереди, среднее время пребывания в системе, дисперсии этих величин, коэффициент простоя прибора и др.

В задании оговаривается требуемая точность оценки заданной характеристики эффективности, что определяет длину реализации или необходимое количество заявок, которые надо сформировать при моделировании.

1. Пример задания

Исходные данные:

1. Количество источников – 2.
2. Характеристики потоков:
 - 2.1. 1-й поток – Эрланга 2-го порядка с интенсивностью порождающего простейшего потока $\Lambda_1=2$.
 - 2.2. 2-ой поток простейший с $\Lambda_2=1$.
3. Буфер – общий на 4 места.
4. Дисциплина записи в буфер – в порядке поступления.
5. Дисциплина выбора из буфера – по приоритету (1-ый поток более приоритетный), а в случае заявок равного приоритета первой выбирается пришедшая раньше.
6. Поток обслуживания – регулярный, причем длительность обслуживания $\tau_{обс}$, меняется в процессе моделирования (при переходе к другой длинной реализации) в заданных пределах при заданном шаге варьирования $\Delta\tau_{обс}=0,2$

$$\tau_1=1 \leq \tau_{обс} \leq \tau_2=2$$

В результате моделирования найти:

1. Вероятности отказов в обслуживании заявок по каждому из источников в зависимости от длительности обслуживания, т.е.

$$P_{отк}(i)=f_{1i}(\tau_{обс})$$

2. Математическое ожидание времени ожидания обслуживания в буфере для заявок об обоих источниках, т.е.

$$M(\tau_{ож})=f_{2i}(\tau_{обс})$$

Требования к точности результатов моделирования:

Относительная погрешность всех оценок – 10% при доверительной вероятности $\alpha=0.85$

2. Этапы разработки алгоритма моделирования

Задание должно быть выполнено с использованием данных объектного типа, однако в дальнейшем изложении не используются объекты, поскольку выбор типа объектов существенно определяет текст программы, и каждый программист в соответствии со своими привязанностями разрабатывает используемые типы объектов (классы).

Вначале же надо хорошо понять существо функционирования моделируемой системы, поэтому все схемы алгоритмов представлены в традиционном стиле.

1 этап – ознакомление и осмысление существа задания и результатов, которые должны быть получены при моделировании. Из приведенного примера задания следует, что требуется построить четыре зависимости, примерный вид которых показан на рис.1. На этом этапе целесообразно изобразить и саму СМО (рис.2). Здесь I_1, I_2 – источники, D_1 – диспетчер записи в буфер (при занятом буфере заявка идет в отказ), D_2 – диспетчер выбора из буфера на прибор, $BUFT$

и *BUFN* – память моментов поступления заявок и номеров источников этих заявок соответственно.

На этом этапе полезно для лучшего понимания функционирования системы нарисовать временные диаграммы ее работы с указанием всех элементов и наиболее характерных ситуаций, которые могут сложиться при работе (см. рис.3).

Цифры в кавычках означают номер источника, которому принадлежит заявка, а перечеркнутые цифры указывают номера заявок, получивших отказ. Такие диаграммы позволяют представить продвижение заявок в системе, а также понять, какие должны быть индикаторы состояния элементов системы, определяющие ее статус.

2 этап – составление общей схемы моделирования.

При этом желательно пользоваться такими именами переменных, которые удовлетворяют правилам языка Pascal, а их аббревиатура соответствует смыслу этих переменных. В частности, рекомендуем следующие имена:

KOL(I) – количество заявок, поступивших в систему от I-го источника;

KOBR(I), *KOTK(I)* – количество обработанных и получивших отказ заявок от I-го источника соответственно;

TPOST(J) – момент поступления заявки от J-го источника;

NMIN – номер источника, от которого пришла заявка раньше;

TНО – момент начала обслуживания заявки прибором;

NAYOB – длительность обслуживания;

TOSV – момент освобождения прибора;

INDBUF – индикатор буфера (количество заявок в буфере);

BUFT(K) – буфер моментов поступления заявок (массив из K элементов);

BUFN(K) – буфер номеров источников, которым принадлежат соответствующие заявки, хранящиеся в BUFT;

LAM(I) – массив интенсивностей входных потоков или потоков обслуживания, если потоки однотипны и их удобно индексировать;

LAMOB – интенсивность потока обслуживания;

TAYI – (или с другими цифрами в конце имени) – параметры других потоков;

DLAM – приращение интенсивностей;

DTAY – приращение параметра потока;

KMIN – длина реализации для достижения заданной точности;

BOTK(I) – вероятность отказа от обслуживания заявки I-го источника;

MTOG(I) – математическое ожидание заявки от I-го источника в буфере;

TOG(I) – общее время ожидания в буфере заявок от I-го источника.

Размеры соответствующих массивов определяются условиями задачи.

При выполнении курсовой работы рекомендуем один из наиболее простых подходов к моделированию подобных систем, а именно **обработку очередного события в активных элементах системы в зависимости от состояния или статуса всех остальных элементов.**

В соответствии с этой методикой за активные элементы описанной выше системы примем прибор и источники заявок. Общая схема моделирования бу-

дет иметь следующий вид (рис.4). Блок определения очередного события (БО-ОС) выбирает наименьшее из трех моментов времени: TPOST(1), TPOST(2), TOSV.

Первый случай соответствует событию «пришла заявка от первого источника», второй случай – событию «пришла заявка от второго источника», а третий – событию «прибор закончил работу (обслужил заявку)». Этим трем случаям соответствуют выходы из блока БООС, помеченные цифрами 1, 2 и 3.

На втором этапе разработки алгоритма надо словами описать назначение каждого из блоков укрупненной схемы алгоритма, представленной на рис.4, т.е. необходимо сформулировать «что надо сделать», пока не задумываясь о том, «как это сделать».

Блоки анализа статуса БАС1, БАС2 и БАС3 в общем случае могут выполнять различные действия по анализу, т.к. эти действия зависят от того, какое событие произошло, и какие изменения в системе надо будет смоделировать как следствие от происшедшего события. Не задумываясь пока о том, насколько схожими будут действия по анализу состояния системы, надо описать (специфицировать) функции БАС1, БАС2 и БАС3.

Блок БАС1 должен обеспечить выполнение действий, являющихся следствием события «пришла заявка от первого источника», а именно:

1. Записать заявку в буфер, если в нем есть место;
2. Отказать в заявке от первого источника, если в буфере нет места;
3. Сформировать следующую заявку первого источника.

При этом действия 1 и 2 взаимно исключают друг друга, а действие 3 должно иметь место всегда.

Блок БАС2 должен обеспечить действия, являющиеся следствием события «Пришла заявка от второго источника», а именно:

1. Записать заявку от второго источника в буфер, если в нём есть место;
2. Отказать в обслуживании заявке от второго источника, если в буфере нет места;
3. Сформировать следующую заявку второго источника.

Первые два действия также исключают друг друга, а третье должно иметь место всегда.

Учитывая, что запись в буфер по условию задачи не зависит от номера источника (бесприоритетна), можно действия БАС1 и БАС2 при дальнейшей детализации алгоритма совместить, а также совместить часть действий в блоках модификации состояния БМС.

Итак, блоки БАС1 и БАС2 анализируют индикатор состояния буфера INDBUF, обеспечивая дальнейшее разветвление на два направления, т. к. для дальнейшего важно знать лишь одно: $INDBUF < 4$ или $INDBUF = 4$.

Блок БАС3 должен обеспечить моделирование следующих событий:

1. Если в буфере есть заявки, взять одну из них на обслуживание в соответствии с дисциплиной выборки, определённой заданием;
2. Если в буфере заявок нет (а произошло событие «Прибор закончил работу»), то прибор необходимо освободить, т. е. вернуться к ситуации, имевшейся в начале моделирования, когда были прогенерированы две заявки, при-

бор был свободен, и решалась судьба заявки, пришедшей раньше. Другими словами, во втором случае никаких изменений, кроме освобождения прибора, моделировать не надо, т. е. можно снова вернуться к блоку БООС, выполняющему функции КАЛЕНДАРЯ событий.

Таким образом, в результате словесного описания функций блоков БАС стало ясным, что необходимо анализировать лишь состояние буфера (точнее, значение переменной INDBUF). Состояние же прибора специально анализировать в БАС не надо, т. к. «Прибор закончил работу», если $TOSV < TPOST$ (NMIN).

3. Определение функций блоков БМС.

Блоки БМС обеспечивают модификацию состояний элементов СМО и моделирование событий, являющихся следствием событий в активных элементах системы – источниках заявок и приборе.

Каждый блок БМС может представлять цепочку действий, состоящих из некоторых типовых, а именно:

1. Запись заявки в буфер;
2. Выборка заявки из буфера на обслуживание;
3. Обслуживание заявки;
4. Генерирование следующей заявки от источника с заданным номером;
5. Фиксация обслуженных заявок, времени ожидания заявок в буфере, числа заявок, получивших отказ и общего числа заявок по каждому из источников;
6. Имитация освобождения прибора.

В соответствии с ранее определёнными функциями блоков БАС1 и БАС2 каждый из них имеет два выхода и определяет по два блока БМС.

Блок БМС₁₁ запускается, если в буфере нет места ($INDBUF = 4$) и должен обеспечить добавление 1 в счётчик КОТК (NMIN), генерирование следующей заявки от источника с номером NMIN, добавление 1 в счётчик КОЛ (NMIN).

Блок БМС₁₂ запускается, если в буфере есть место ($INDBUF < 4$), и обеспечивает запись в буфер, генерирование следующей заявки от источника с номером NMIN, добавление 1 в счётчик КОЛ (NMIN).

Блоки БМС₂₁ и БМС₂₂ обеспечивают аналогичные действия.

Блок БМС₃₁ запускается, если в буфере есть заявки ($INDBUF <> 0$), и обеспечивает выборку заявки из буфера на обслуживание, фиксацию номера источника этой заявки, фиксацию времени ожидания обслуживания этой заявкой, модификацию буфера (сдвиг информации в нём после выборки заявки и изменение значения INDBUF), а также формирование ТНО и TOSV.

Блок БМС₃₂ запускается, если прибор освободился и буфер пуст. В этом случае обслуживается (без предварительной записи в буфер) заявка со временем поступления TPOST (NMIN), а затем генерируется следующая заявка от того же источника. На рис. 5 изображен укрупненно фрагмент алгоритма моделирования, иллюстрирующий взаимодействие блоков БООС, БАС и БМС. При этом блоки БАС₁ и БАС₂ объединены в один, а БМС₁₂ и БМС₁₁ содержат дейст-

вия, которые также можно объединить (генерацию следующей заявки и формирование $KOL(NMIN)$). Кроме того, на рис. 5 блоки $БМС_{21}$ и $БМС_{22}$ не изображены, т.к. они отличаются от $БМС_{11}$ и $БМС_{12}$ соответственно лишь номером заявки, судьба которой в этих блоках определяется, и этот номер источника задан переменной $NMIN$. После разработки главной части укрупненного алгоритма (рис. 5) можно приступить к детализации описания действий в каждом блоке рис. 5 с учетом принятых обозначений переменных.

4. Детализация блоков БМС.

Проще всего детализируется блок генерации следующей заявки.

Прогенерировать заявку – значит сформировать момент поступления ее в систему, зная момент поступления предыдущей заявки и закон распределения интервалов между заявками в потоке по каждому из источников. Поэтому рассмотрим основные формулы, определяющие интервалы между событиями, происходящими в системе. Эти интервалы являются непрерывными случайными величинами с заданным законом распределения.

1. В случае регулярного потока событий интервалы между событиями постоянны и $TAY = \text{const}$.

2. Если поток событий равномерный, то он характеризуется двумя константами $TAY1$ и $TAY2$, а само случайное значение интервала вычисляется по формуле:

$TAY = TAY1 + (TAY2 - TAY1) * R$, где R – случайное число в интервале от 0 до 1, формируемое генератором случайных чисел (функция `Random` в Паскале).

3. В случае простейшего потока интервал между событиями потока вычисляется по формуле

$TAY = -1 / LAMBDA * \ln(R)$, где $R = \text{Random}$, а $LAMBDA$ – интенсивность простейшего потока событий, т.е. среднее число событий в единицу времени.

4. В случае потока Эрланга n -го порядка интервалы между событиями вычисляются как сумма n интервалов простейшего потока с интенсивностью порождающего его простейшего потока $LAMBDA$. Вычисление интервала проводится по формуле

$TAY = -1 / LAMBDA * (\ln(R1 * R2 * \dots * Rn))$, где $R1 - Rn$ – случайные числа в интервале от 0 до 1, сформированные генератором `Random`.

Входной информацией в блоке генерации является известный номер источника, заявку от которого необходимо сформировать ($NMIN$). Значение $NMIN$ позволяет в зависимости от заданных законов распределения сформировать $TPOST(NMIN) = TPOST(NMIN) + TAY(NMIN)$, что иллюстрируется рисунком 6.

Входными данными блока записи в буфер (рис. 7) являются индикатор буфера $INDBUF$ и ранее сформированные значения моментов поступления заявок $TPOST(1)$ и $TPOST(2)$, причем в буфер записывается $TPOST(NMIN)$.

Выходными данными являются $INDBUF$ и пополненные значения $BUFT$ и $BUFN$.

Для выборки из буфера заявки на обслуживание необходимо вначале определить номер места в буфере $NOMOB$, откуда выбирается заявка в соответствии с дисциплиной выборки, заданной условиями задачи.

Входными данными при этом являются $NOMBUF = 1$, а выходными – измененные значения $NOMBUF$ (см. рис. 8). При этом происходит выборка с первого места (ближе к прибору) буфера ($NOMBUF = 1$), если на втором, третьем или четвертом месте нет более приоритетной заявки.

Функции выборки из буфера и последующей его модификации реализуются действиями, описанными в алгоритме на рис. 9, причем, вначале предполагается, что заявка будет выбираться с первого места буфера ($NOMBUF = 1$). Если в буфере была всего одна заявка, то после ее выборки сдвиг буфера не требуется, а номер источника заявки, выбираемой на обслуживание $NOMOB$, определяется по содержимому $BUFN(1)$.

Входными данными в блоке выборки и модификации буфера являются $INDBUF$, $BUFT$ и $BUFN$, а выходными – номер выбранной заявки на обслуживание $NOMOB$, выбранное время ее поступления TOB и номер места $NOMBUF$ в буфере, откуда выбирается заявка.

На рис. 10 представлен алгоритм сдвига заявок в буфере левее выбранной, т.е. начиная с заявки с номером $NOMBUF + 1$ и кончая заявкой с номером $INDBUF$.

Блок обслуживания формирует время ожидания по тому источнику, заявка которого взята на обслуживание ($NOMOB$), а также новое время освобождения. Кроме того, в счетчик обслуженных заявок добавляется единица (рис. 11).

Блок «Формирование $TOSV$ » в $БМС_{32}$ отличается от блока обслуживания только тем, что в нем не формируется TOG , т.к. в этом случае $TOSV < TPOST(NMIN)$ и блок «Формирование $TOSV$ » будет иметь вид, приведенный на рис. 12.

Детализация блока обработки результатов одной реализации (рис. 4) сводится к формированию вероятностей отказов $BOTK(I)$ и оценки математического ожидания времени ожидания $MTOG$ заявок каждого источника (рис. 13).

Перед началом моделирования необходимо в алгоритме задать начальные значения данным общим для всего алгоритма, а именно, задать значения интенсивностей входных потоков $LAM(1)$ и $LAM(2)$, значение интервала обслуживания $TAY1$, а также $TAY2$ и $DTAY$. Кроме того, требуется задать значение $KMIN$, исходя из заданных характеристик точности.

Данным же $KOL, KOTK, KOBR, TOSV, TOG, INDBUF$ в начале каждой новой реализации присвоить нулевые начальные значения. Кроме того, к моменту первого обращения к блоку БООС в каждой реализации необходимо сформировать по одной заявке от каждого источника.

5 . Рекомендации по организации вывода промежуточных данных в алгоритме.

Вывод промежуточных результатов моделирования является одним из важнейших моментов, предопределяющих успех на этапе отладки алгоритма, поэтому очень важно продумать вопрос о том, что и где (в каком месте алгоритма) надо выводить на печать .

Основной критерий правильности решения этого вопроса- достаточность выведенной информации для обнаружения и локализации логических ошибок в алгоритме. Рекомендуется выводить такие данные и в таких местах, которые бы позволили проверить все ветви алгоритма. Выводить же их надо с признаком , определяющим место вывода. Желательно выводить при этом какую-либо поясняющую дополнительную информацию. Форматы выводимых данных должны обеспечить экономное расходование бумаги или объем памяти для выходного файла на диске, в частности , неразумно выводить по одному значению в строке.

Применительно к рассмотренному в методических указаниях примеру минимальный объем сводится к выводу всех переменных при входе в блок БООС. Это позволит определить , по каким ветвям проходит алгоритм и правильно ли формировались при этом переменные. После отладки программы рядом с операторами вывода можно поставить символ " Комментарий " , после чего будут выводиться лишь окончательные результаты *ВОТК* , *МТОГ* , *ТАУОВ* .

6. Определение длины реализации.

В рассмотренном типовом задании все числовые характеристики (интенсивности входных потоков , параметры закона обслуживания , число мест в памяти заявок и т.д.) задаются преподавателем . Но одна величина , а именно *KMIN* , определяется в результате пробного моделирования . При этом можно использовать следующие формулы (вывод см. в приложении) :

-при вычислении оценки вероятности какого-либо события , например , вероятности отказа

$$KMIN_1 = (t^2_{\alpha}(1-p)) / \delta^2 p;$$

-при вычислении математического ожидания $M(x)$

$$KMIN_2 = (t^2_{\alpha} \delta^2(x)) / \delta^2 M^2(x);$$

-при вычислении дисперсии какой-либо величины

$$KMIN_3 = (2 \delta^3 t_{\alpha}) \delta.$$

Здесь приняты следующие обозначения :

- $KMIN_1$ -наименьшая длина реализации (число заявок от источника с наименее высокой плотностью потока);

- $KMIN_2$ - наименьшее число слагаемых для получения оценок S^2 (вместо σ^2) и среднего значения X вместо $M(x)$;
- $KMIN_3$ - наименьшее число слагаемых для получения оценки дисперсии какой-либо величины;
- δ - относительная точность , с которой требуется вычислить оценку методом статистического моделирования при заданной доверительной вероятности α , определяющей значение t_α , входящее в формулы для длины реализации $KMIN$. При заданном значении α значение t_α^2 можно взять из нижеприведенной таблицы.

α	t_α^2	α	t_α^2
0.8	1.64	0.96	4.21
0.85	2.08	0.97	4.49
0.9	2.71	0.98	5.43
0.95	3.84	0.99	6.61
		0.999	10.9

Процедура итерационного определения значения $KMIN_1$ состоит в следующем :

1. Задается небольшое значение $KMIN_1^{(1)}$ и проводится моделирование с длиной реализации $KMIN_1^{(1)}$.
2. Вычисляется оценка $p^{(1)}$ искомой вероятности, и уточняется по вышеприведенной формуле необходимое значение $KMIN_1^{(2)}$.
3. Проводится моделирование с длиной реализации $KMIN_1^{(2)}$, и вычисляется оценка $p^{(2)}$, по которой уточняется значение $KMIN_1 = KMIN_1^{(3)}$ и т. д.

Последовательность получаемых оценок $p^{(i)}$ в этой итеративной процедуре должна сходиться к искомому значению p . Аналогично методом пробного моделирования определяется необходимое для достижения заданной точности δ моделирования при заданной доверительной вероятности α (или t_α) значение $KMIN_2$ и $KMIN_3$.

В настоящей работе рекомендуется упрощенный способ оценки значения $KMIN$.

Вычисляется значение оцениваемой величины в зависимости от количества прогенерированных заявок по наименее интенсивному источнику. Эти вычисления проводятся с определенным шагом по числу прогенерированных заявок (например, через каждые 100 заявок). Результаты выводятся на экран в графическом режиме. При достаточно большом числе заявок эти результаты стабилизируются. То значение, при котором они практически перестают меняться, и есть значение $KMIN$. Это отчетливо видно на графиках выводимых величин.

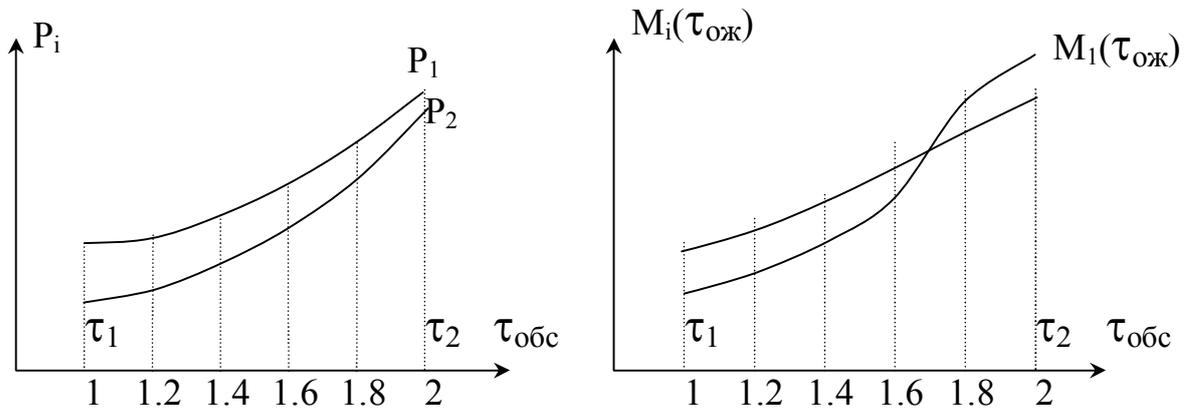


Рис.1

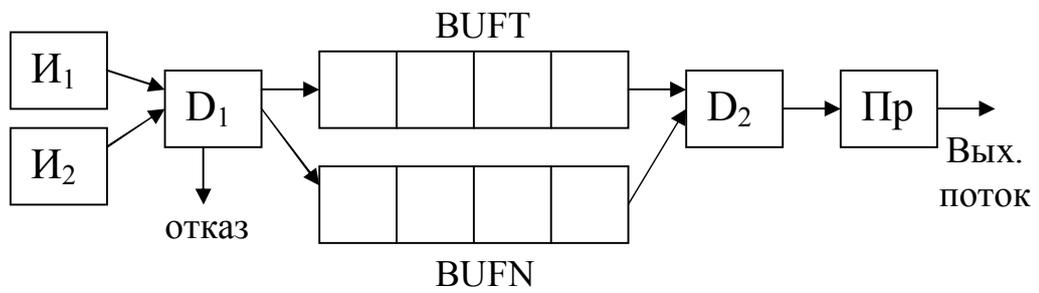


Рис.2

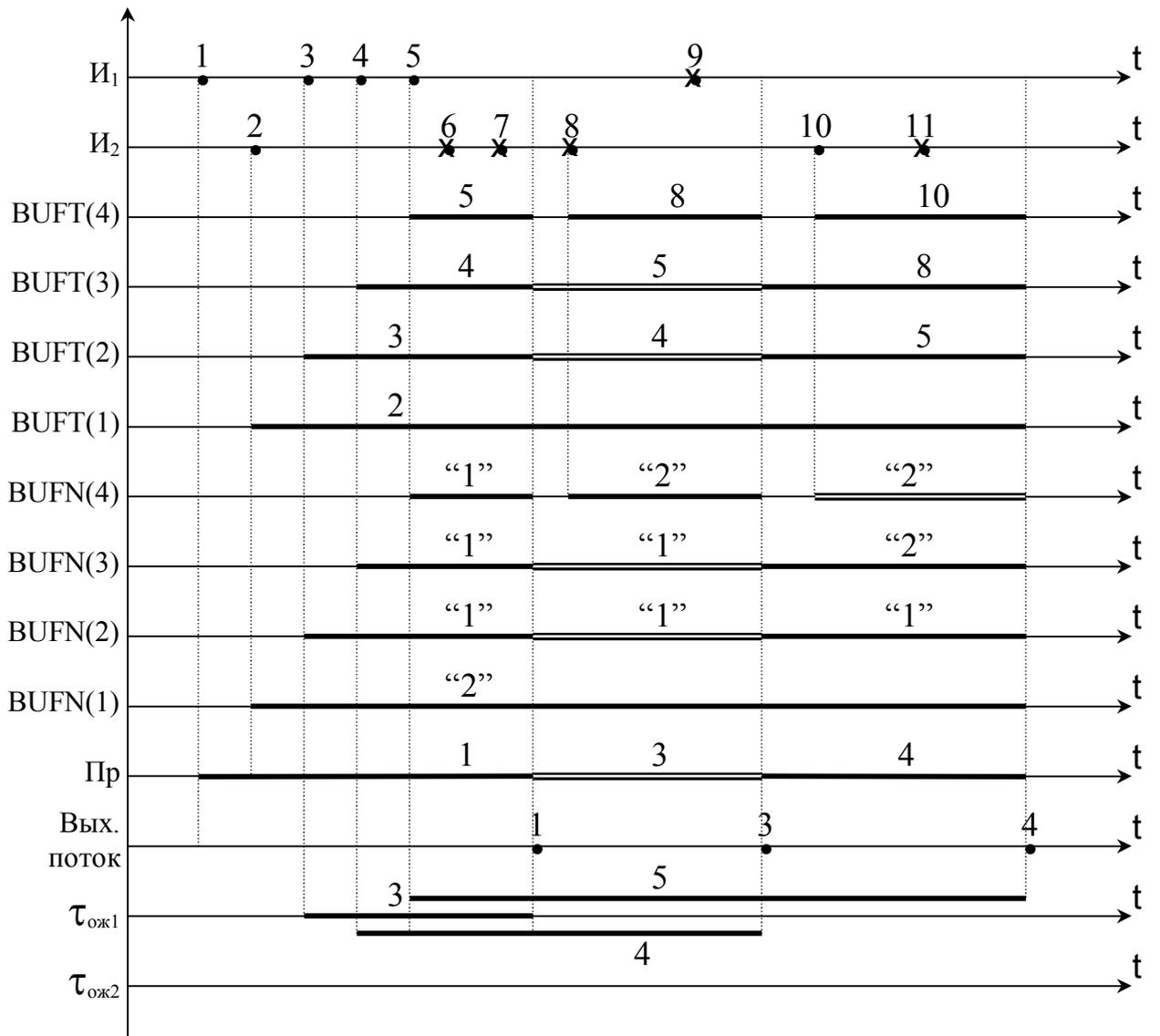


Рис.3

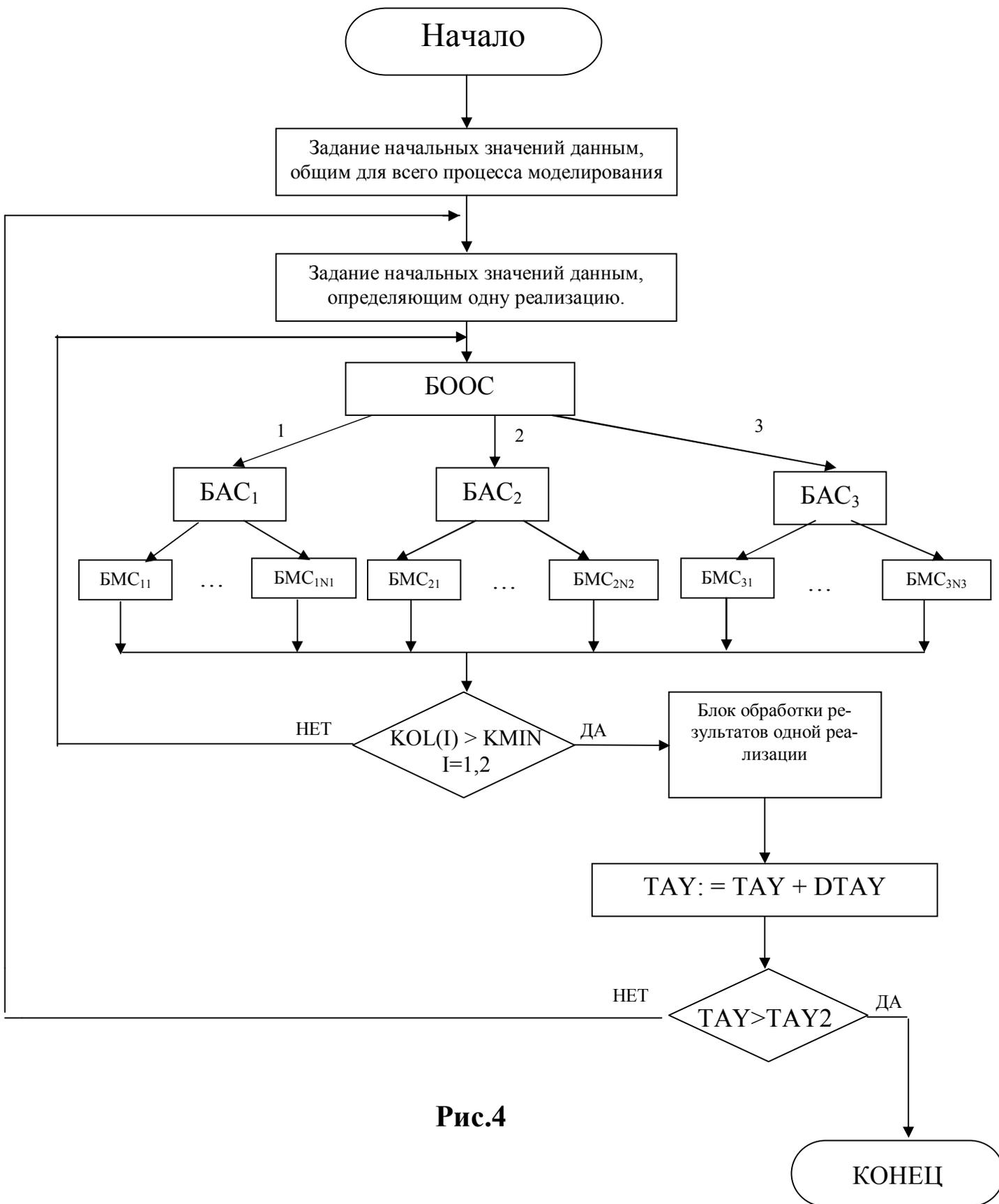
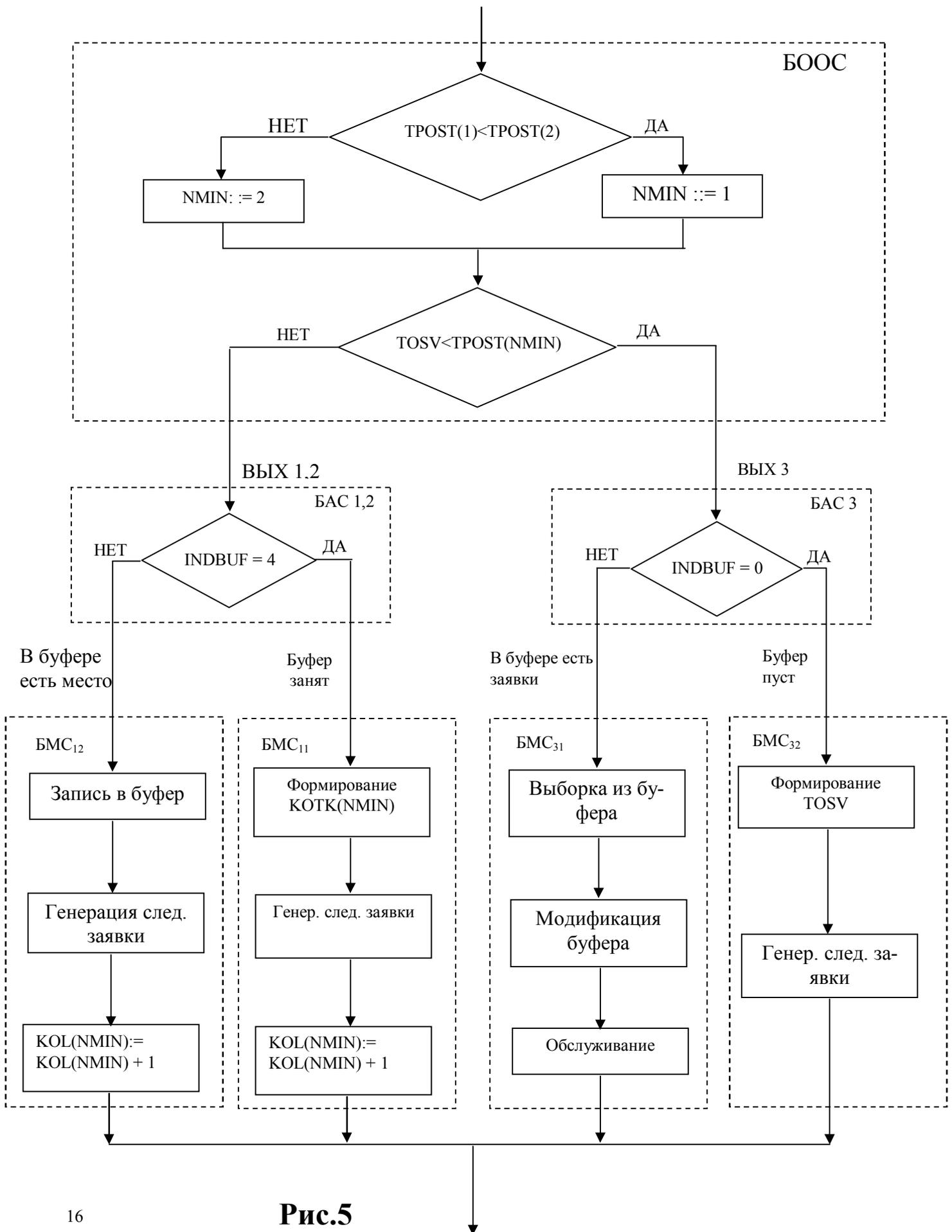


Рис.4



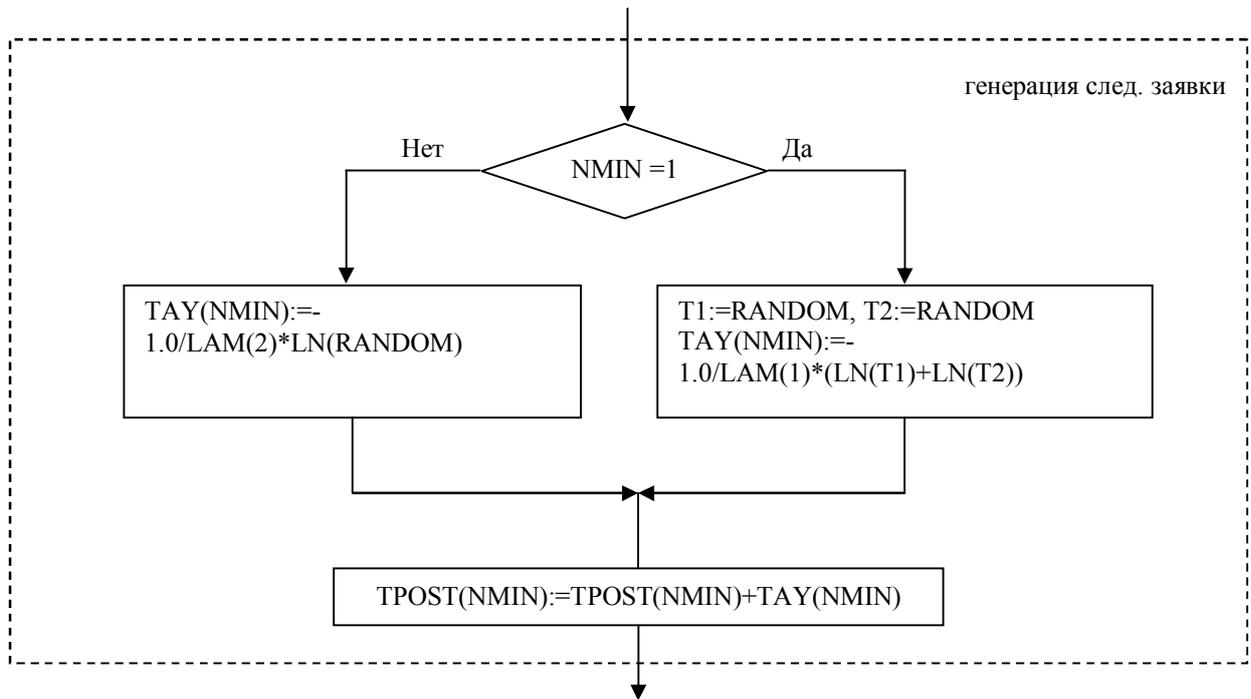


Рис.6

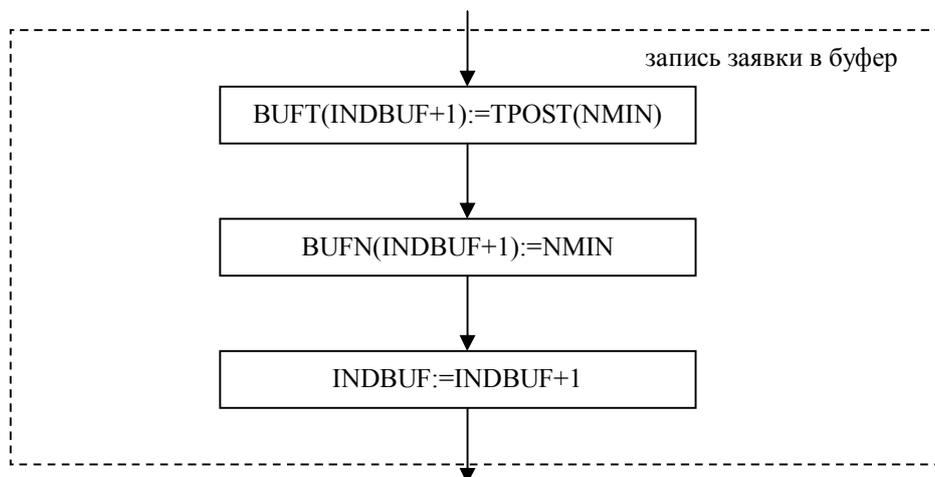


Рис.7

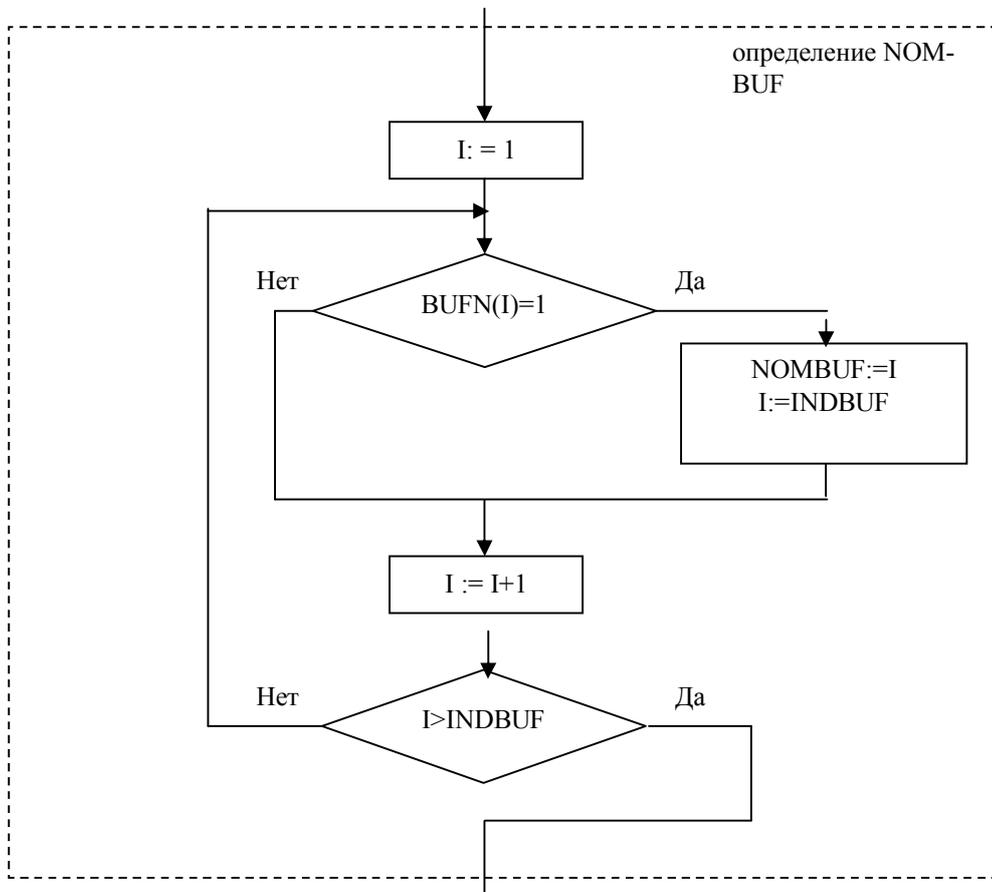


Рис.8

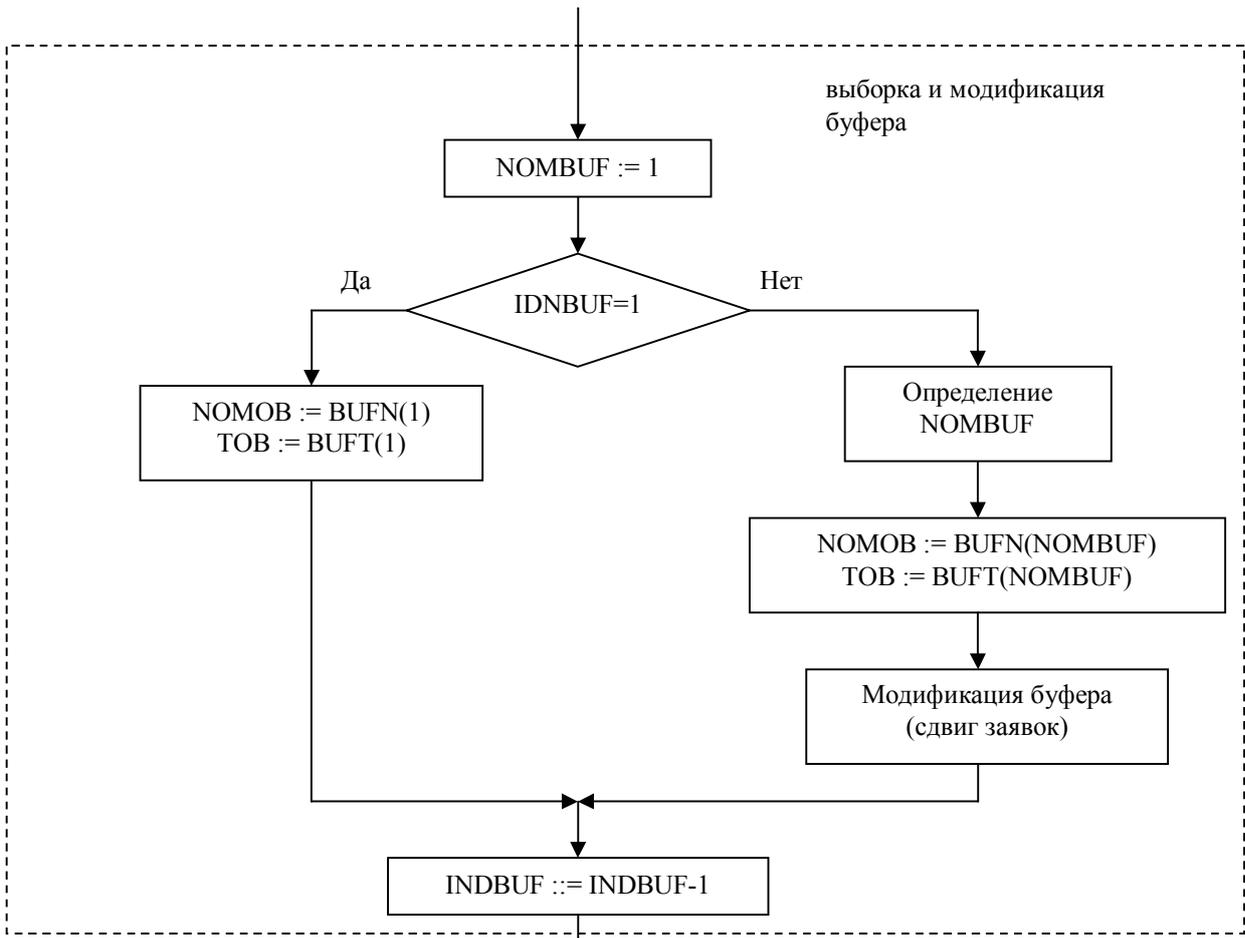


Рис.9

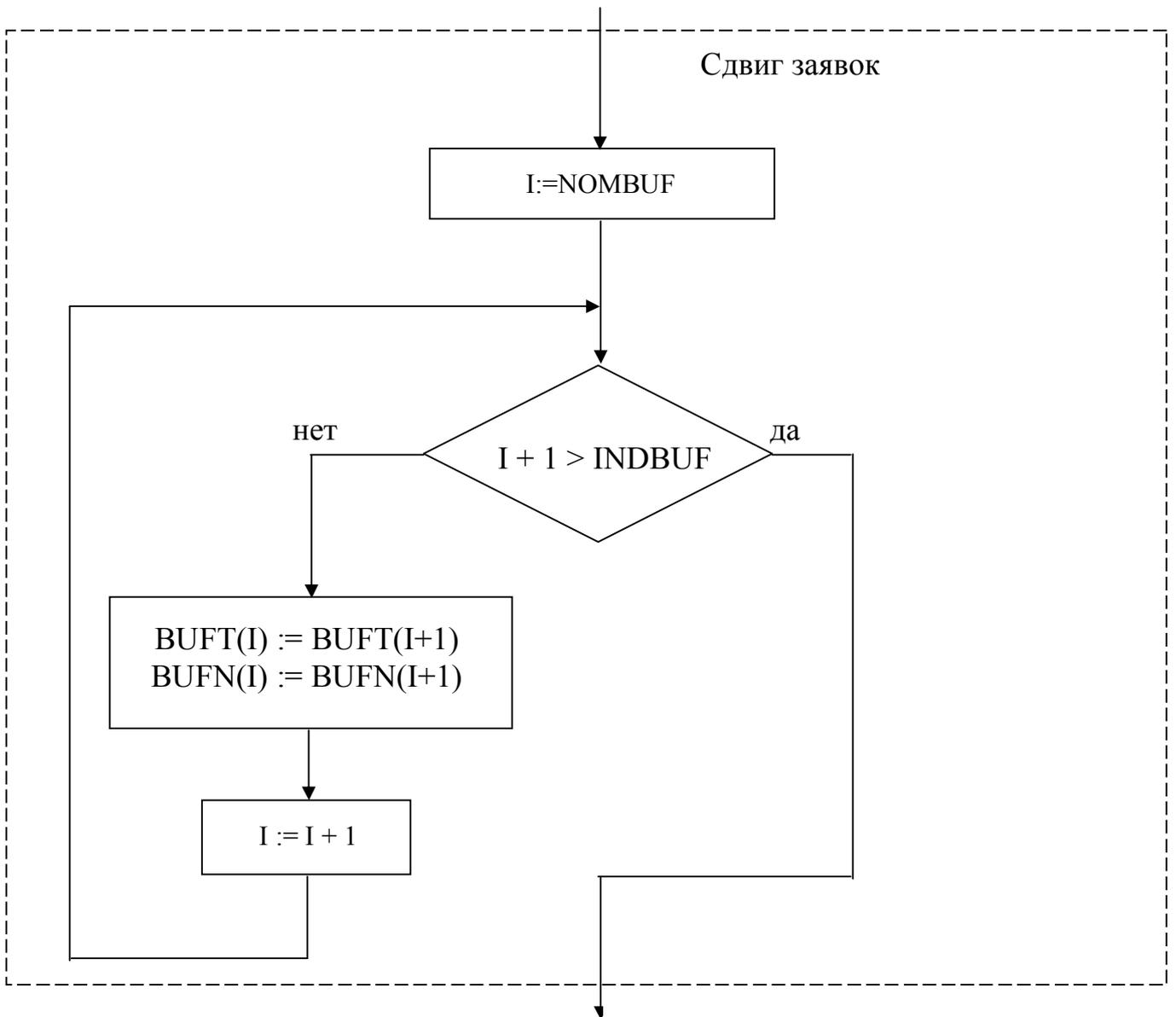


Рис.10

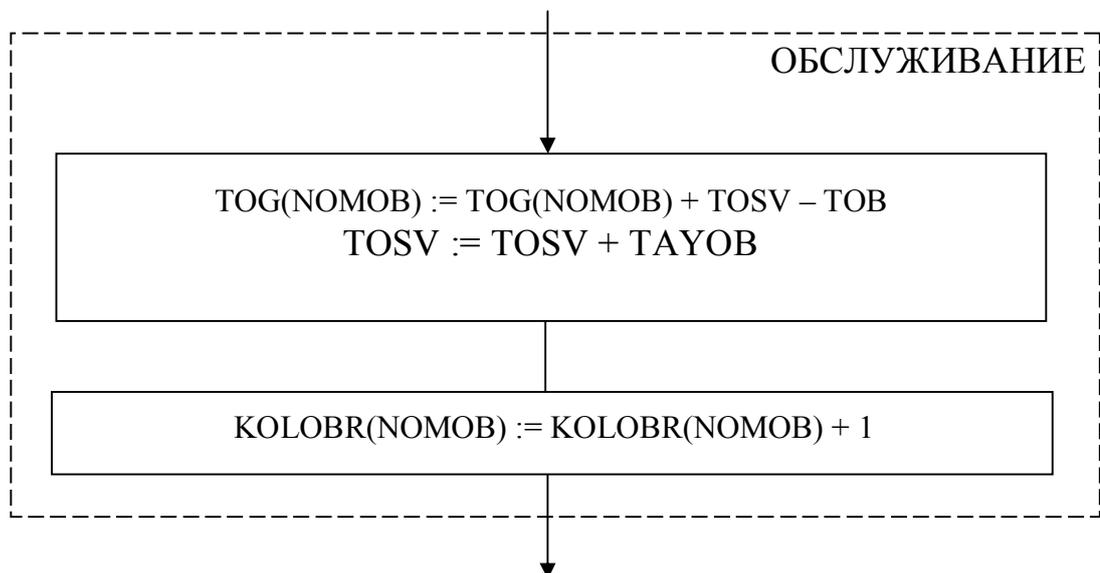


Рис.11

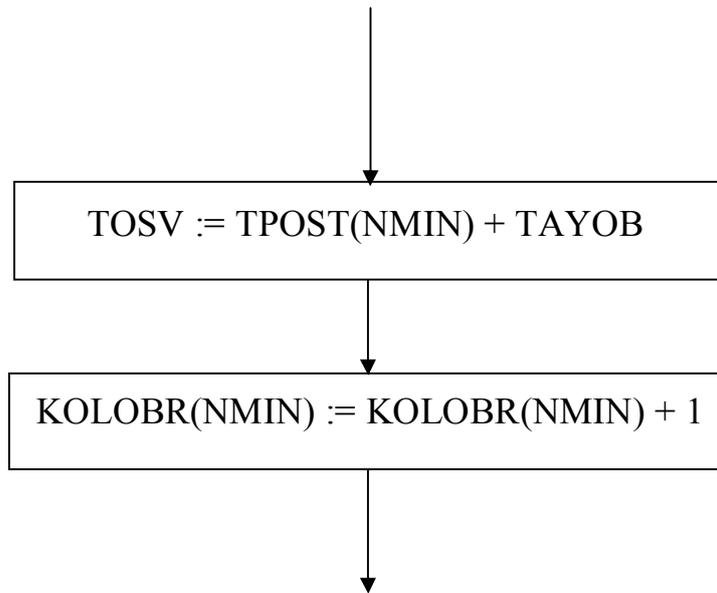


Рис.12



Рис.13

7. Состав программного продукта курсовой работы.

Предлагаемая курсовая работа является первым относительно большим программным продуктом, разрабатываемым в процессе обучения на кафедре ИУС. Поэтому здесь особенно важна дисциплина разработки. При разработке программист самостоятельно принимает решения по использованию различных структур данных и организации взаимодействия работы различных частей проекта. Поэтому никакие шаблоны для разработки не навязываются, но разработка должна проводиться при некоторых небольших ограничениях.

1. Программный продукт должен состоять из нескольких модулей.

Можно рекомендовать не менее трех вспомогательных модулей, например, функциональный, модуль интерфейса и графический модуль, не считая главного модуля.

2. Всю графику исполнять, используя лишь модуль Graph. Нельзя использовать библиотеку Turbo Vision, поскольку одна из основных целей курсовой работы – научиться разрабатывать собственные типы объектов.

3. Помимо файлов текстов на языке Pascal разработчик представляет файл Readme.doc, который должен содержать информацию, необходимую и достаточную для эксплуатации программного продукта любым пользователем, знакомым только с существом работы системы массового обслуживания.

4. Все модули разрабатываются **только на переменных объектного типа.**

5. При использовании объектов желательно применить по необходимости различные возможности объектного программирования, предусмотренного версией языка Turbo Pascal 7.0, а именно, инкапсуляцию, полиморфизм, переопределение методов, виртуальные методы, методы с формальными параметрами, динамические объекты и пр. При этом не надо искусственно усложнять взаимодействие типов объектов, но и не делать объекты формально, т.е. даже без полей данных, как некоторые нерадивые студенты пытаются делать.

8. Рекомендации по разработке модулей.

Функциональный модуль.

Функциональный модуль содержит описания всех типов используемых в нем объектов, например, объекты типа 'Прибор', 'Источник' или 'Буфер' или какие-либо другие объекты, в зависимости от замысла разбивки на объекты всей моделируемой системы. При этом могут быть использованы некоторые вспомогательные модули для моделирования функционирования предложенной системы. Но не надо усложнять реализацию самого функционирования системы.

Можно эту часть работы выполнить в виде одного модуля.

Методы функционального модуля должны обеспечить формирование результатов одной длинной реализации, т.е. зависимость интересующих нас величин (вероятностей отказов, математического ожидания времени пребывания в системе и пр.) от количества заявок, поступивших от самого ‘медленного’ источника, т.е. источника с минимальной интенсивностью потока заявок. Эти данные в процессе моделирования сразу же в графическом режиме надо выводить на экран, с оцифровкой по осям координат. Именно на основании поведения этих кривых и выбирается необходимое число реализаций KMIN.

Кроме того, этот модуль должен обеспечить фиксацию окончательных результатов моделирования при варьировании основного параметра системы, который изменяется при переходе к другой длинной реализации. Эти результаты целесообразно записывать либо в какой-либо текстовый файл либо в массив записей, из которого затем графический модуль обеспечит построение графиков зависимостей выходных характеристик моделируемой системы от варьируемого параметра системы.

Модуль интерфейса.

Интерфейсный модуль содержит описание объектов, позволяющих формировать пользовательского меню с выпадающим подменю. В качестве основных пунктов меню можно принять следующие: ‘Установка параметров’, ‘Моделирование’, ‘Результаты’, ‘Помощь’. У каждого пункта основного меню должно быть по 2-3 пункта подменю. Например, для ‘Установки параметров’ можно предусмотреть все устанавливаемые параметры системы. Для пункта ‘Моделирование’ можно предусмотреть подпункты ‘С отображением графиков’ зависимостей параметров от числа заявок и ‘Без отображения процесса моделирования’. В меню ‘Результаты’ можно предусмотреть вывод окончательных зависимостей в виде таблиц либо в виде графиков и предусмотреть вывод любой заданной величины. В пункте ‘Помощь’ предусмотреть вывод основной информации инструкции пользователя, которая содержится в файле Readme.doc.

Модуль графики.

Модуль графики содержит основные объекты, необходимые для вывода в процессе моделирования зависимостей интересующих параметров от числа заявок. При этом информация для этих графиков (например, вероятности отказов) получается и сразу же выводится на график. Не надо формировать большие массивы и только потом их выводить. Графики строятся в режиме ‘Online’ в самом процессе моделирования.

Кроме того, этот модуль должен обеспечить вывод результирующих зависимостей интересующих параметров от варьируемой в задаче величины. Эти

результаты должны быть зафиксированы в конце каждой реализации в соответствующих массивах или файлах , и только по желанию пользователя в конце окончания всего процесса моделирования они из этих массивов должны выводиться на графики с помощью процедур этого графического модуля

9. Требования к исходным текстам и файлу Readme.doc.

1. Все описания типов объектов (поля данных, методы и их формальные параметры) должны быть прокомментированы прямо в описательной части текстов программ, т.е. там должно быть кратко описано назначение полей , параметров и методов.

Сами же тексты комментировать не требуется.

2. В файле Readme.doc должно быть задание на разработку , инструкция программисту и инструкция оператору.

Инструкция программисту должна содержать подробный состав всего программного продукта , назначение и содержимое всех файлов , особенности их работы в той или иной среде и особенности их применения при использовании среды Turbo Pascal 7.0

(например, особенности каталогов, в которых должны находиться те или иные необходимые компоненты программного продукта или элементов интегрированной среды).

Инструкция оператора должна содержать информацию о том, как правильно работать с данным программным продуктом, а именно, подробное описание пользовательского интерфейса , всех пунктов и подпунктов меню , как работать с клавиатурой или мышью и как в нормальном режиме на то или иное действие оператора должна реагировать программа. Кроме того, необходимо описать нештатные ситуации и как должен вести себя оператор в этих случаях.

10. Порядок разработки.

1. Детально изучить и понять функционирование системы.

2. Продумать и описать все типы используемых объектов в функциональном модуле.

3. Разработать функциональный модуль, задавая параметры в тексте иллюстрирующей программы , и отладить этот модуль, задавая небольшое значение KMIN , выводя и детально анализируя значения формируемых величин с целью отладки самой логики алгоритма.

4. Разработать часть модуля пользовательского интерфейса, позволяющего запустить программу и просмотреть результаты в выходном текстовом файле.

5. Доработать графический модуль с целью просмотра в графике зависимость оцениваемых величин от длины реализации.

6. Выбрать пробным моделированием значение KMIN.

7. Запустить программу целиком, варьируя исходные параметры системы и еще раз оценив качественное поведение окончательных зависимостей от этих параметров.

8. Написать файл Readme.doc.

Приложение.

Оценка точности определения характеристик моделируемой системы методом статистического моделирования.

При статистическом моделировании приходится иметь дело со случайными величинами. Однако нас интересуют вполне определенные (детерминированные) величины. В описанной курсовой работе надо вычислить детерминированные характеристики системы, такие как Р-вероятность отказа обработки заявок, М(Т)-математическое ожидание времени пребывания заявки в системе, К-коэффициент занятости прибора и т.д.

Эти детерминированные величины не могут быть получены методом статистического моделирования, а вместо них получаются случайные их значения или их оценки. Так, вместо вероятности Р(А) какого-либо события получается частота события А :

$P^* = m/N$, где m- число наступления события А, а N- общее число испытаний.

Вместо математического ожидания какой-либо случайной величины М(Х) получается ее среднее значение :

$$\bar{X} = 1/N \sum_{i=1}^N X_i \quad (1)$$

В общем случае при желании вычислить какую-либо детерминированную величину В мы получаем ее случайную оценку В1. Про В1 нельзя сказать, что она отличается от искомой величины в точности на величину ошибки ε , а можно лишь говорить про вероятность этой ошибки :

$$p(|B - B1| < \varepsilon) = \alpha \quad (2)$$

Где ε - абсолютная погрешность полученной оценки, а α - доверительная вероятность оценки или ее достоверность.

Цель дальнейшего изложения – получить формулы для необходимой длины реализации КМІN при заданных значениях абсолютной (или относительной) ошибок оцениваемых величин и заданной доверительной вероятности α .

Пусть событие А наступает с вероятностью Р и пусть при наступлении этого события дискретная случайная величина ξ принимает значение 1, а если событие не наступило – значение 0.

Тогда математическое ожидание этой дискретной случайной величины $M(\xi)$ и ее дисперсия $D(\xi)$ равны соответственно:

$$M(\xi) = P \cdot 1 + (1-P) \cdot 0 = P \quad (3)$$

$$D(\xi) = (1-M(\xi))^2 \cdot P + (0-M(\xi))^2 \cdot (1-P) = P \cdot (1-P)$$

Нас интересует математическое ожидание частоты наступления события A и дисперсия этой частоты $D(m/N)$.

Частота P^* наступления события A равна:

$$P^* = m/N = 1/N \sum_{i=1}^N \xi_i$$

$$m = \sum_{i=1}^N \xi_i \quad (4)$$

Отсюда получаем:

$$M(m/N) = 1/N \cdot M\left(\sum_{i=1}^N \xi_i\right) = 1/N \cdot \sum_{i=1}^N M(\xi_i) = 1/N \cdot M(\xi) \cdot N = M(\xi) = P \quad (5)$$

$$D(m/N) = D\left(1/N \sum_{i=1}^N \xi_i\right) = 1/N \cdot D\left(\sum_{i=1}^N \xi_i\right) = 1/N \cdot N \cdot D(\xi) = D(\xi) = P(1-P)/N$$

Таким образом, математическое ожидание частоты наступления события A и ее среднее квадратичное отклонение σ равны:

$$M(m/N) = P \quad (6)$$

$$D(m/n) = P(1-P)/N \quad (7)$$

$$\sigma = \sqrt{P(1-P)/N} \quad (8)$$

Если нас интересует математическое ожидание среднего значения случайной величины, то оно определяется в соответствии с формулами теории вероятностей:

$$\bar{X} = 1/N \sum_{i=1}^N X_i \quad (9)$$

$$M(\bar{X}) = 1/N \cdot M(X_i) = M(X) \quad (10)$$

$$D(\bar{X}) = 1/N \cdot D\left(\sum_{i=1}^N X_i\right) = D(X)/N \quad (11)$$

$$\sigma(\bar{X}) = \sigma(X) / \sqrt{N} \quad (12)$$

В соответствии с центральной предельной теоремой “При большом числе испытаний их средний результат распределен по нормальному закону”.

В нашем случае частота P^* и среднее значение \bar{X} случайной величины X распределены по нормальному закону с математическим ожиданием и дисперсией, определенными соответственно полученными выше формулами (6), (8) и (10),

(12) соответственно.

После этих предварительных рассмотрений получим формулу для значения $KMIN$ для получения оценки вероятности отказов при заданной абсолютной погрешности ε этой оценки и доверительной вероятности α .

Из формулы для доверительной вероятности следует :

$$P(|P^* - P| < \varepsilon) = \alpha = P(-\varepsilon \leq P^* - P \leq \varepsilon) = P\left(-\frac{\varepsilon}{\sigma_{p^*}} \leq t \leq \frac{\varepsilon}{\sigma_{p^*}}\right) \quad (13)$$

Если от переменной перейти к центрированной и нормированной переменной t , связанной с переменной зависимостью:

$$t = \frac{P^* - P}{\zeta_{p^*}} \quad (14)$$

то выражение (12) превратится в :

$$\frac{1}{\sqrt{2\pi}} \int_{-\frac{\varepsilon}{\sigma_{p^*}}}^{\frac{\varepsilon}{\sigma_{p^*}}} e^{-\frac{t^2}{2}} dt = 2F\left(\frac{\varepsilon}{\sigma_{p^*}}\right) = \alpha, \quad (15)$$

где F-функция Лапласа, которая протабулирована и имеет вид:

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\varepsilon}{\sigma_{p^*}}}^{\frac{\varepsilon}{\sigma_{p^*}}} e^{-\frac{t^2}{2}} dt \quad (16)$$

Из (15) следует :

$$\frac{\varepsilon}{\sigma_{p^*}} = F^{-1}\left(\frac{\alpha}{2}\right) = t_\alpha \quad \text{или} \quad \frac{\varepsilon\sqrt{N}}{\sqrt{p(p-1)}} = t_\alpha, \quad (17)$$

откуда получаем длину реализации, т.е. необходимое число заявок от самого “медленного” источника, которое надо прогенерировать для достижения заданной относительной погрешности δ получения оценки вероятности отказа при заданной доверительной вероятности α

$$N = \text{KMIN} = \frac{t_\alpha^2 p(p-1)}{\varepsilon^2} = \frac{t_\alpha^2 (1-p)}{\delta^2 p} \quad (18)$$

Таблица зависимости t_α^2 от α была приведена выше.

Аналогично получаются формулы для оценки необходимого значения длины реализации KMIN при получении оценки математического ожидания $M(X)$ какой-либо величины, заменяя его на среднее значение случайной величины X .

При этом используются параметры нормального закона, представленные формулами (10) и (12):

$$\text{KMIN} = \frac{t_\alpha^2 \sigma^2(x)}{\varepsilon^2} = \frac{t_\alpha^2 \sigma^2(x)}{\delta^2 M^2(x)} \quad (19)$$

где $\varepsilon = \delta M(x)$

Поскольку ни $M(x)$ ни $\sigma^2(x)$ неизвестны, то вместо них надо использовать их оценки, получаемые как случайные значения, т.е. \bar{X} и статистическую дисперсию $s^2 = 1/N \sum_{i=1}^N (\bar{X} - X)^2$.