

Практическая работа «Имитационное моделирование на GPSS»

Цель работы: знакомство с системой имитационного моделирования GPSS World, запуск системы, изучение блоков GENERATE и TERMINATE, освоение способов управления продолжительностью моделирования, разработка простейших GPSS-моделей

Порядок выполнения работы

1. Изучить методические указания.
2. Построить в среде GPSS модель
3. Провести эксперименты с имитационной моделью, получить результаты и проанализировать их.
4. Разработать модель в соответствии с индивидуальным заданием.

Требования к оформлению отчета

1. Название и цель работы, программное обеспечение.
2. Постановка задачи.
3. Описание модели на GPSS.
4. План и анализ результатов вычислительных экспериментов с моделью.
5. Вместе с отчетом должны быть файлы с моделями

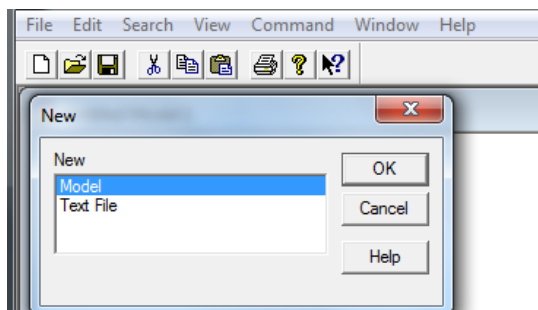
Методические указания по разработке моделей на GPSS

Задание № 1. Запуск первой программы на GPSS World

Ход работы:

GPSS World является программой имитационного моделирования и применяется в системах массового обслуживания, моделирования систем (магазины, бензозаправки, метро, разгрузка и загрузка груза и т.д.)

1. Запустить программу:
2. Выполнить: File / New
3. Выбрать: Model, нажать Ок



4. Написать простую программу:

GENERATE 1 ; Время генерирования одной транзакции

TERMINATE 1 ; Удаляем одну транзакцию

START 10 ; время моделирования 10 ед. времени (минут)

Операторы вводятся не вручную, а командой Edit/ Insert GPSS Block:

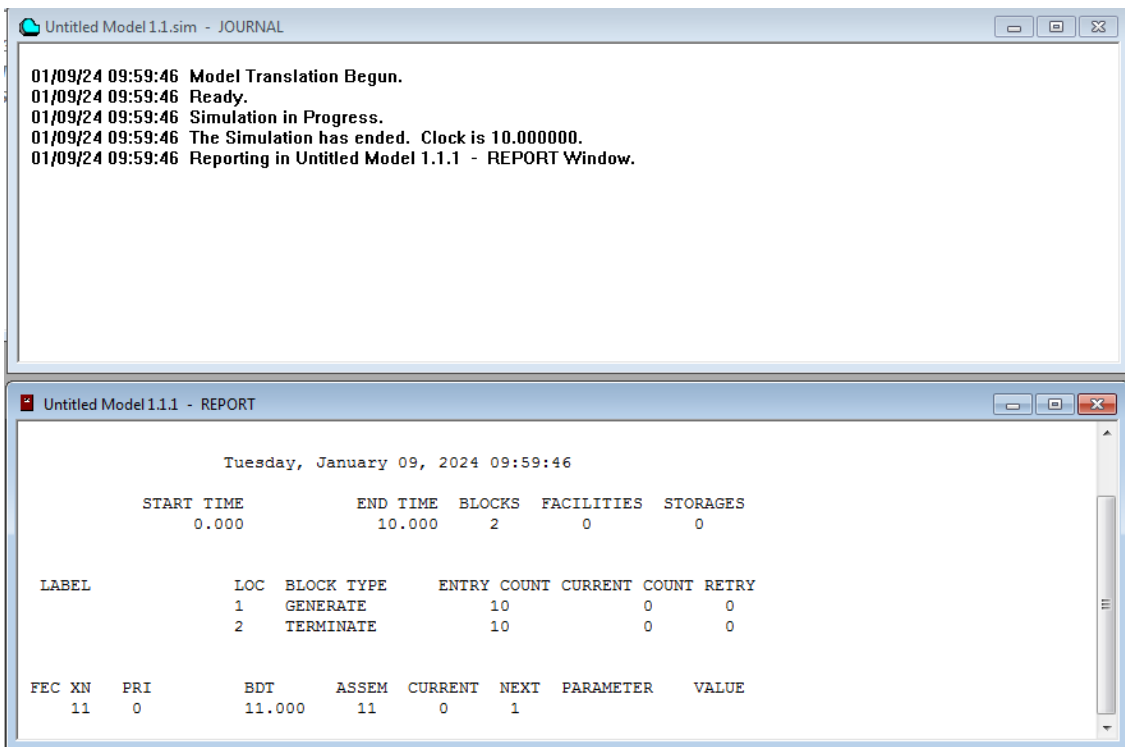


Insert GPSS Block into Model Object		
ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Выбрать в таблице нужный оператор и ввести его значение

5. Запустить программу, для этого перейти на вкладку Command / Create Simulation

В результате отобразятся два окна: журнал ошибок и листинг отчёта.



The screenshot shows two windows from the GPSS simulation software. The top window is titled 'Untitled Model 1.1.sim - JOURNAL' and contains the following text:

```
01/09/24 09:59:46 Model Translation Begun.
01/09/24 09:59:46 Ready.
01/09/24 09:59:46 Simulation in Progress.
01/09/24 09:59:46 The Simulation has ended. Clock is 10.000000.
01/09/24 09:59:46 Reporting in Untitled Model 1.1.1 - REPORT Window.
```

The bottom window is titled 'Untitled Model 1.1.1 - REPORT' and contains the following report:

```
Tuesday, January 09, 2024 09:59:46

START TIME      END TIME  BLOCKS  FACILITIES  STORAGES
0.000           10.000    2        0           0

LABEL          LOC  BLOCK TYPE  ENTRY COUNT  CURRENT COUNT  RETRY
1              1    GENERATE    10           0           0
2              2    TERMINATE   10           0           0

FEC XN  PRI      BDT    ASSEM  CURRENT  NEXT  PARAMETER  VALUE
11      0      11.000  11     0       1     
```

№ 2. Среднее время прихода клиента к железнодорожной кассе для покупки билета составляет 12 ± 3 минут. Время оформления билета кассиром составляет 5 ± 2 . Составить программу и промоделировать обслуживания кассиром 10-ти клиентов.

Ход работы:

Моделирование потока клиентов будем выполнять с помощью оператора GENERATE (Генерировать).

GENERATE [A],[B],[C],[D], [E]

A — средний интервал времени;

B — половина значения интервала в соответствии с допуском (равномерное распределение);

Учитывая, что интервал времени прибытия клиентов колеблется в пределах 12 ± 3 мин, то оператор будет иметь вид:

```
GENERATE 12,3
```

В поле операнда A указывается средний интервал времени между прибытием в магазин двух идущих один за другим покупателей (требований, транзактов). В нашем примере он составляет 12 мин.

В поле операнда B дано отклонение времени прихода покупателей от среднего. В нашем примере это отклонение составляет 3 мин. Покупатель, пришедший в магазин, сначала встает в очередь к кассиру, если она есть.

Клиент, пришедший к железнодорожной кассе, сначала встает в очередь к кассиру, если она есть. Это можно промоделировать оператором QUEUE (Очередь), который только в совокупности с соответствующим оператором DEPART (Выйти) собирает статистическую информацию о работе моделируемой очереди.

```
QUEUE A
```

В нашем примере оператор QUEUE будет выглядеть так:

```
QUEUE QKASSA
```

где QKASSA - имя очереди (Очередь в кассу).

Следуя логике, покупатель может выйти из очереди только тогда, когда освободится кассир (канал обслуживания). Для этого вводится оператор SEIZE, который определяет занятость канала обслуживания, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так: SEIZE KASSA

В поле операнда A дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя KASSA (Кассир).

Выход клиента из очереди фиксируется оператором DEPART с соответствующим названием очереди. В нашем примере это будет выглядеть так: DEPART QKASSA

Должно быть промоделировано время пребывания покупателя, непосредственно обслуживаемого кассиром. Это время в нашем примере составляет 5 ± 2 мин. Для моделирования этого процесса используется оператор ADVANCE (Задержать), который в нашей задаче будет выглядеть так: ADVANCE 5,2

После обслуживания кассиром клиент отправляется к продавцу за получением оплаченного товара. Однако перед этим системе должно быть послано сообщение об

освобождении канала обслуживания. Это делается с помощью оператора RELEASE, который в нашей задаче записывается так: RELEASE KASSIR
GENERATE 12,3 ; время генерирования прихода клиентов
QUEUE QKASSA ; занять очередь
SEIZE KASSA ;занять кассу
DEPART QKASSA ; освободить очередь
ADVANCE 5,2 ; обслуживание кассиром
RELEASE KASSA ; покинуть кассу
TERMINATE 1 ; удаление транзакта
START 10

Таким образом программный код имеет вид:

```
GENERATE 12,3 ; время генерирования прихода клиентов
QUEUE QKASSA ; занять очередь
SEIZE KASSA ;занять кассу
DEPART QKASSA ; освободить очередь
ADVANCE 5,2 ; обслуживание кассиром
RELEASE KASSA ; покинуть кассу
TERMINATE 1 ; удаление транзакта
START 10
```

В результате отображается журнал и листинг отчета

№ 3. Построить гистограмму, которая определяет справится ли кассир с потоком клиентов или образуется очередь. Клиенты для получения денег приходят на кассу каждые 8 ± 4 минут. Обслуживание у кассира занимает 7 ± 2 . Программа, при в 100 единиц времени моделирования, будет иметь вид:

```
GENERATE 8,4 ; время генерирования прихода клиентов
QUEUE KASSIRU ; занять очередь
SEIZE KASSA ;занять кассу
DEPART KASSIRU ; освободить очередь
ADVANCE 7,2 ; обслуживание кассиром
RELEASE KASSA ; покинуть кассу
TERMINATE 1 ; удаление транзакта
START 100
```

Чтобы построить гистограмму, необходимо в начало исходной программы добавить блок

```
WAIT QTABLE KASSIRU,0,2,15
```

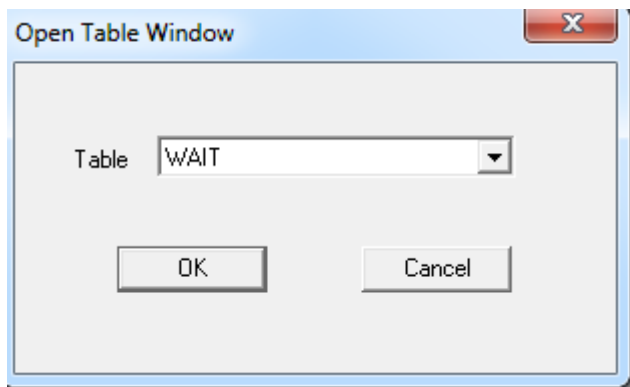
Блок QTABLE означает построение гистограммы очереди, 2 — это шаг гистограммы, 15 — это количество интервалов.

Запустить симуляцию программы: Command /Create Simulation.

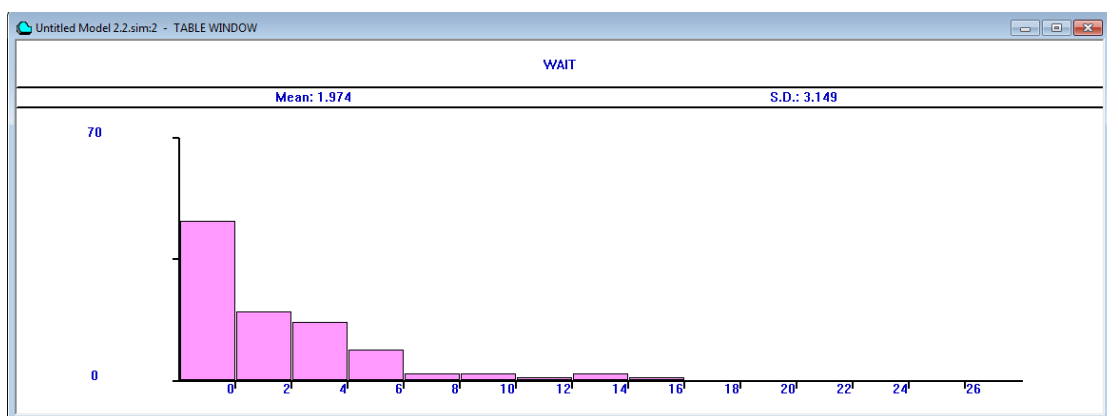
В результате выводится отчёт программы

Построить гистограмму: Window / Simulation Window / Table Window

Появляется окно Open Table Window, выбираем WAIT и нажимаем ОК



В итоге, получаем гистограмму, из которой видно, что кассир не справляется с потоком клиентов и образуются очереди.



№ 4. Построить график очереди клиентов, взяв за основу данные предыдущей задачи
WAIT QTABLE KASSIRU,0,2,15

GENERATE 8,4 ; время генерирования прихода клиентов

QUEUE KASSIRU ; занять очередь

SEIZE KASSA ;занять кассу

DEPART KASSIRU ; освободить очередь

ADVANCE 7,2 ; обслуживание кассиром

RELEASE KASSA ; покинуть кассу

TERMINATE 1 ; удаление транзакта

START 100

В блоке START 100 поменяем значение на START 1

Блок с оператором QTABLE для построения графика не обязателен.

Выполнить команду: Window -> Simulation Window -> Plot Window

В результате появится окно Edit Plot Window, в котором

–label название очереди на графике

–Q\$KASSIRU — текущая длина очереди с именем KASSIRU

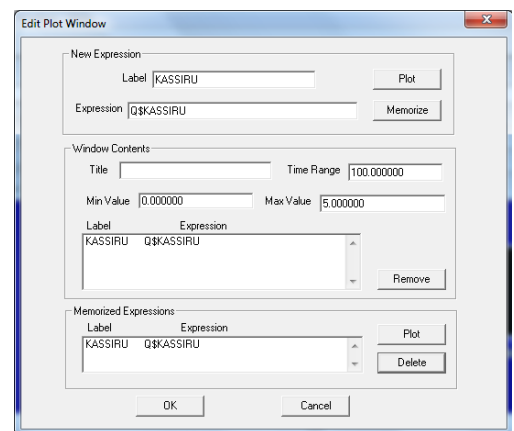
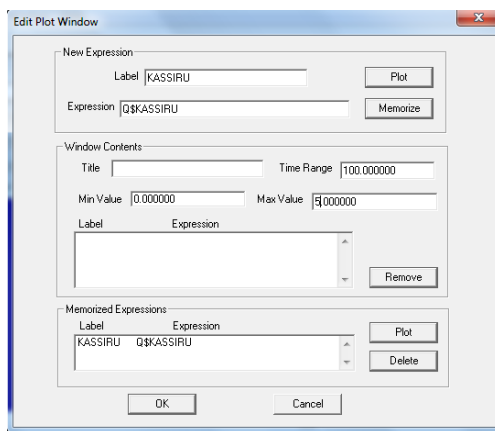
–Title — общее название графика

–Time Range — время моделирование

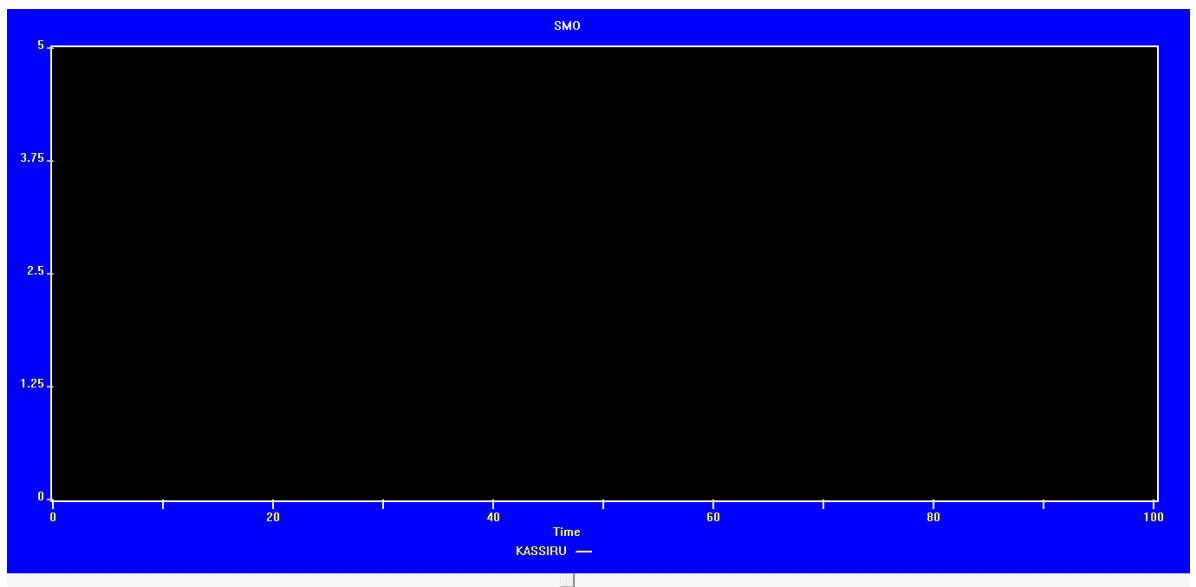
–Min value и Max value — минимальное и максимальное значение очереди по оси Y

(вертикальная ось). Max value = 5

Нажать Plot и Memorize, после этого значения появляются в полях Memorized Expression и Window Contents, нажимаем Ок



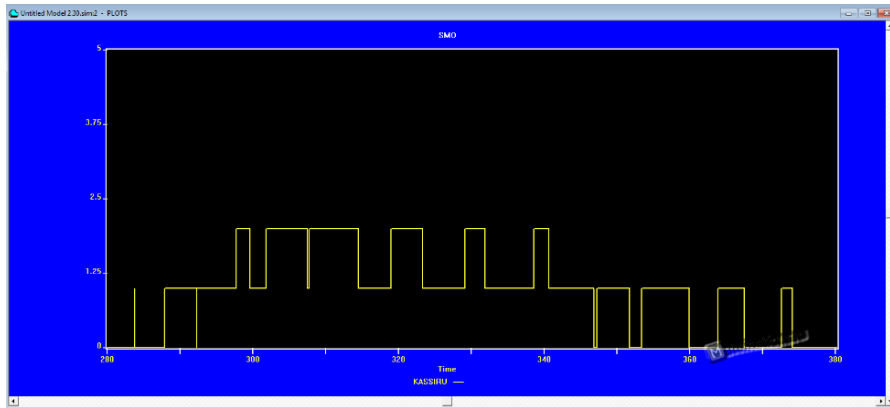
Отображается пустая область для графика



Перейти на вкладку Command / START

Задать значение 100 и нажать Ок

В результате отобразится график очереди клиентов к кассиру и как видно из графика очередь больше одного человека за длительный промежуток времени, что является отказом в одноканальной СМО.



№ 5. Построить имитационную модель работы магазина, который имеет один кассовый аппарат и одного продавца. Известны следующие параметры функционирования магазина:

- поток покупателей (требований), приходящих в магазин за покупками, равномерный;
 - интервал времени прибытия покупателей колеблется в пределах $9,5 \pm 0,8$ мин;
 - время пребывания покупателей у кассового аппарата составляет $2,3 \pm 0,7$ мин.
- После этого покупатели подходят к продавцу для получения товара;
- время, потраченное на обслуживание покупателей продавцом, составляет $10 \pm 1,4$ мин.
- Требуется определить параметры функционирования магазина:

- коэффициент загрузки кассира;
- коэффициент загрузки продавца;
- максимальное, среднее и текущее число покупателей в каждой очереди;
- среднее время обслуживания в каждом канале обслуживания;
- среднее время нахождения покупателя в каждой очереди и др.

Ход работы:

1. Создать модель. Выполнить команду: File/New/ выбрать Model
2. Появится окно модели

Добавить оператор QTABLE с меткой t_prod для сбора информации и построения соответствующей гистограммы функционирования очереди под именем Ocher_prod.

```
t_prod QTABLE Ocher_prod,0,2,32
```

Моделирование потока покупателей будем выполнять с помощью оператора GENERATE (Генерировать). В нашем примере он будет выглядеть так:

```
GENERATE [A],[B],[C],[D], [E]
```

A — средний интервал времени;

B — половина значения интервала в соответствии с допуском (равномерное распределение);

Т.к. интервал времени прибытия покупателей колеблется в пределах $9,5 \pm 0,8$ мин,
то

GENERATE 9.5,0.8

В поле операнда A указывается средний интервал времени между прибытием в магазин двух идущих один за другим покупателей (требований, транзактов). В нашем примере он составляет 9,5 мин.

В поле операнда B дано отклонение времени прихода покупателей от среднего. В нашем примере это отклонение составляет 0,8 мин. Покупатель, пришедший в магазин, сначала встает в очередь к кассиру, если она есть.

Покупатель, пришедший в магазин, сначала встает в очередь к кассиру, если она есть. Это можно промоделировать оператором QUEUE (Очередь), который только в совокупности с соответствующим оператором DEPART (Выйти) собирает статистическую информацию о работе моделируемой очереди.

QUEUE A

В нашем примере оператор QUEUE будет выглядеть так:

QUEUE Ocher_kassa

где Ocher_kassa - имя очереди (Очередь в кассу).

Следуя логике, покупатель может выйти из очереди только тогда, когда освободится кассир (канал обслуживания). Для этого вводится оператор SEIZE, который определяет занятость канала обслуживания, и при его освобождении очередное требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так: SEIZE Kassir

В поле операнда A дается символьное или числовое имя канала обслуживания. В нашей задаче каналу дано имя Kassir (Кассир).

Выход покупателя из очереди в кассу фиксируется оператором DEPART с соответствующим названием очереди. В нашем примере это будет выглядеть так: DEPART Ocher_kassa

Должно быть промоделировано время пребывания покупателя, непосредственно обслуживаемого кассиром. Это время в нашем примере составляет $2,3 \pm 0,7$ мин. Для моделирования этого процесса используется оператор ADVANCE (Задержать), который в нашей задаче будет выглядеть так: ADVANCE 2.3,0.7

После обслуживания кассиром покупатель отправляется к продавцу за получением оплаченного товара. Однако перед этим системе должно быть послано сообщение об освобождении канала обслуживания. Это делается с помощью оператора RELEASE, который в нашей задаче записывается так: RELEASE Kassir

Парные операторы QUEUE и DEPART для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам SEIZE и RELEASE.

После обслуживания в кассе покупатель направляется к продавцу – следующему каналу обслуживания. Процесс моделирования этой цепи аналогичен только что описанному. И в нашем примере он может быть представлен, например, в таком виде:

```
QUEUE Ocher_prod
SEIZE Prodavec
DEPART Ocher_prod
ADVANCE 10,1.4
RELEASE Prodavec
```

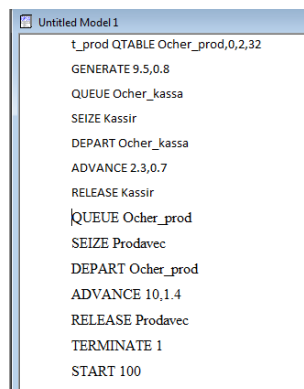
После обслуживания продавцом (каналом обслуживания) покупатель (требование) покидает систему. Это действие может быть представлено оператором TERMINATE (Завершить): TERMINATE 1

Число 1 - означает, что систему обслуживания – магазин – покупатели покидают по одному.

Завершающим оператором в нашей задаче является управляющая команда START (Начать), позволяющая начать моделирование: START 100

Число 100 - показывает, с каким числом покупателей будет моделироваться система работы магазина.

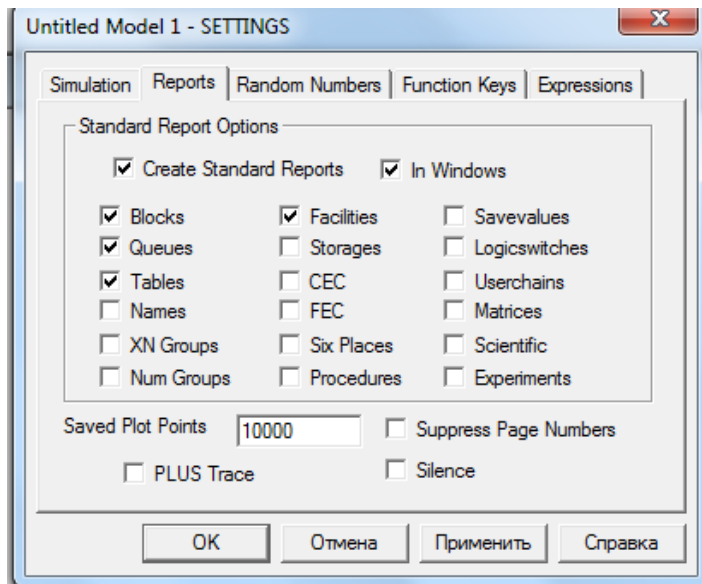
В результате программа имеет вид



```
Untitled Model 1
t_prod QTABLE Ocher_prod,0,2,32
GENERATE 9.5,0.8
QUEUE Ocher_kassa
SEIZE Kassir
DEPART Ocher_kassa
ADVANCE 2.3,0.7
RELEASE Kassir
QUEUE Ocher_prod
SEIZE Prodavec
DEPART Ocher_prod
ADVANCE 10,1.4
RELEASE Prodavec
TERMINATE 1
START 100
```

3. Подготовка к моделированию системы

Выполнить: Edit (Правка)/ Settings (Установки). Появится диалоговое окно SETTINGS для данной модели, в котором можно установить нужные выходные данные. Для нашего примера это может выглядеть так, как представлено на рисунке



Наличие галочки в окошках говорит о том, что эта информация будет выведена в окне результатов моделирования. В нашем примере будет выведена информация для следующих объектов:

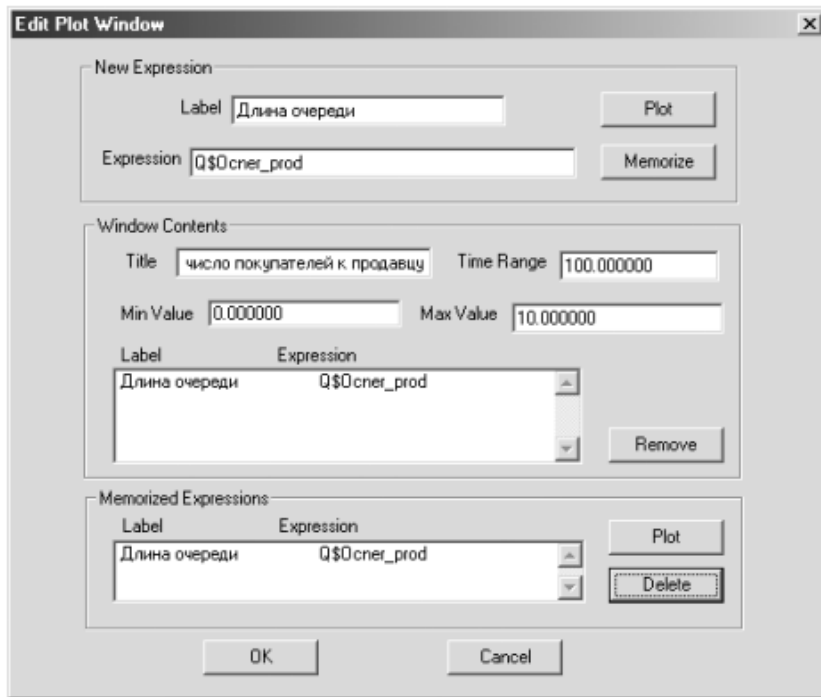
- Blocks (Блоки);
- Queues (Очереди);
- Tables (Таблицы/гистограммы);
- Facilities (Каналы обслуживания).

После создания имитационной модели необходимо оттранслировать и запустить на выполнение. Для этого:

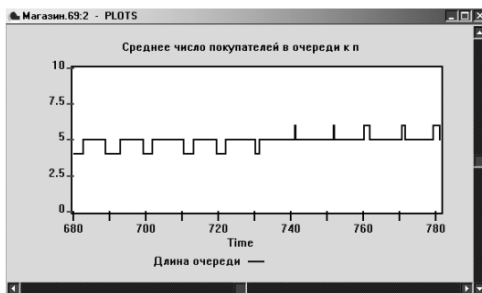
- выполнить Command / Create Simulation (Создать выполняемую модель)

Если управляющая команда START есть в модели, то исходная имитационная модель после трансляции, если в ней нет ошибок, начнет выполняться, появятся окна JOURNAL. Будет выполняться то число прогонов, которое указано в поле операнда команды START, т.е. 100.

Настроить внешний вид отображения графика, для этого выполнить команду Window/Simulation Window / Plot Window (Окно графика) Появится диалоговое окно Edit Plot Window (Окно редактирования графика), которое необходимо соответствующим образом заполнить. В поле Title: «Среднее число покупателей к продавцу», остальные параметры, как на рисунке. Ввести параметры и нажать Plot, Memorize



Выполнить команду: Command /Start / введите в диалоговом окне Start Command число посетителей магазина 1000, и щелкните по кнопке ОК.



При выводе графика на передний окно REPORT с результатами моделирования переместится на задний план. Для просмотра окна REPORT щелкните по нему мышью. Оно вновь перейдет на передний план и будет выглядеть так, как показано

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1016.324	12	2	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1		GENERATE	107	0	0
2		QUEUE	107	0	0
3		SEIZE	107	0	0
4		DEPART	107	0	0
5		ADVANCE	107	0	0
6		RELEASE	107	0	0
7		QUEUE	107	6	0
8		SEIZE	101	1	0
9		DEPART	100	0	0
10		ADVANCE	100	0	0
11		RELEASE	100	0	0
12		TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KASSIR	107	0.244	2.318	1	0	0	0	0	0
PRODAVEC	101	0.987	9.929	1	101	0	0	0	6

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OCHER_PROD	7	107	2	3.607	34.260	34.913	0
OCHER_KASSA	1	0	107	0.000	0.000	0.000	0

TABLE	MEAN	STD. DEV.	RANGE	RETRY FREQUENCY	CUM. %
T_PROD	34.338	17.466		0	

В верхней строке окна REPORT (Отчет) указываются:

- START TIME (Начальное время) – 0.000;

- END TIME (Время окончания) – 1016.324;
- BLOCKS (Число блоков) – 12;
- FACILITIES (Число каналов обслуживания) – 2;
- STORAGES (Число накопителей) – 0.

Ниже перечисляются блоки модели и количество входов в них требований (покупателей). При этом каждый блок имеет свой числовой номер. Еще ниже указываются результаты моделирования каналов обслуживания под назначенными нами именами KASSIR и PRODAVEC соответственно:

- ENTRIES (Число входов) – 107, 101;
- UTIL. (Коэффициент использования) – 0.244, 0.987;
- AVE. TIME (Среднее время обслуживания) – 2.318, 9.929;
- AVAIL. (Доступность) – 1, 1;
- OWNER (Возможное число входов) – 0, 101;
- PEND – 0, 0;
- INTER – 0, 0;
- RETRY (Повтор) – 0, 0;
- DELAY (Отказано) – 0, 6.

Еще ниже указываются результаты моделирования каждой очереди под присвоенными нами именами OCHER_PROD и OCHER_KASSA соответственно:

- MAX (Максимальное содержание) – 7 и 1;
- CONT. (Текущее содержание) – 7 и 0;
- ENTRY (Число входов) – 107 и 107;
- ENTRY(0) (Число нулевых входов) – 2 и 107;
- AVE.CONT. (Среднее число входов) – 3.607 и 0.000;
- AVE.TIME (Среднее время) – 34.260 и 0.000;
- AVE.(–0) – 34.913 и 0.000;
- RETRY – 0 и 0.

Еще ниже указываются результаты моделирования для построения по табличным данным гистограммы T_PROD функционирования очереди под именем OCHER_PROD:

- MEAN (Средняя) – 34.338;
- STD.DEV. (Среднее квадратическое отклонение) – 17.466;
- RANGE (Область);
- RETRY – 0;
- FREQUENCY (Частота);
- CUM.% (Суммарный процент)

При наличии оператора `t_prod` QTABLE `Ocher_prod,0,2,32` можно вывести соответствующую гистограмму. Для этого выполнить команду: `Window / Simulation Window` (Окно моделирования) / `Table Window` (Окно гистограммы). Появится диалоговое окно `Open Table Window` (Открыть окно гистограммы).

Окно гистограммы очереди к продавцу в модели «Магазин» имеет вид:

