

В общем, есть два способа добавления блоков GPSS:

- Написать его вручную
- В меню Edit выбрать Insert GPSS Block и выбрать необходимый блок


Первый способ заметно быстрее, но у второго есть преимущество: отображаются описания всех параметров.

Если вкратце, то моделирование необходимо начинать с описания входящего потока объектов или *транзактов*. В случае с моделированием транспортного потока в роли транзактов будут выступать автомобили, при моделировании цеха по обработке деталей транзактами будут эти детали.

#### Создание и удаление транзактов

Для генерации транзактов используется блок Generate. Первым параметром является время между созданием двух транзактов. Если указать только этот параметр, то транзакты будут генерироваться с постоянным интервалом времени. Например, каждые 15 секунд.

Enter Block Information

 **GENERATE**

GENERATE - Create XN for future entry.

A:  Intergeneration time.

B:  Halfrange or Function Modifier.

C:  Start delay time.

D:  Creation limit.

E:  Priority.

F:

G:

H:

Label:

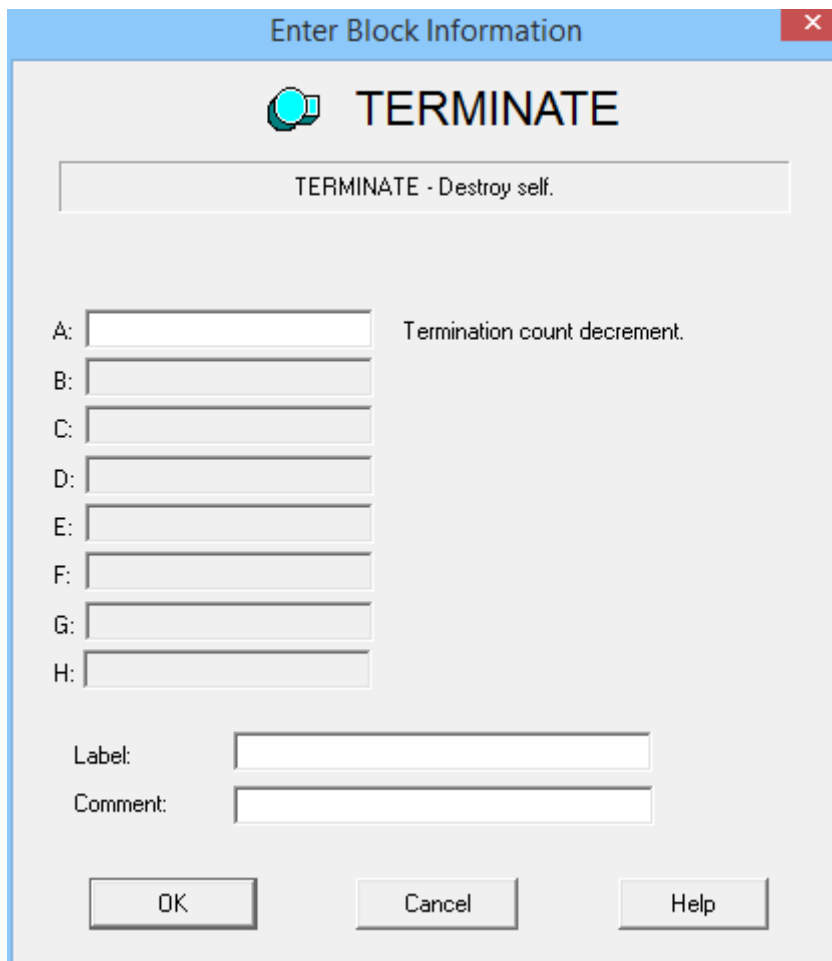
Comment:

OK Cancel Help


Вторым параметром является разброс интервала времени относительно первого параметра. Например, если указать первым параметром 10, а вторым – 2, то время генерации транзакта будет находиться в промежутке от 8 до 12.

Третьим параметром является время задержки при генерации стартового транзакта (по умолчанию равно 0). Четвертый – количество генерируемых транзактов (по умолчанию бесконечно). Пятый – приоритет транзактов (по умолчанию = 0).

После того, как транзакт пройдет по системе, его необходимо удалить. Для удаления используется блок Terminate.



Enter Block Information

 **TERMINATE**

TERMINATE - Destroy self.

A:  Termination count decrement.

B:

C:

D:

E:

F:

G:

H:

Label:

Comment:

OK Cancel Help

Параметр у него один (и не обязательный) – уменьшение счетчика завершений. Про него – чуть ниже.

Запуск симуляции

Допустим, у нас есть следующий код:

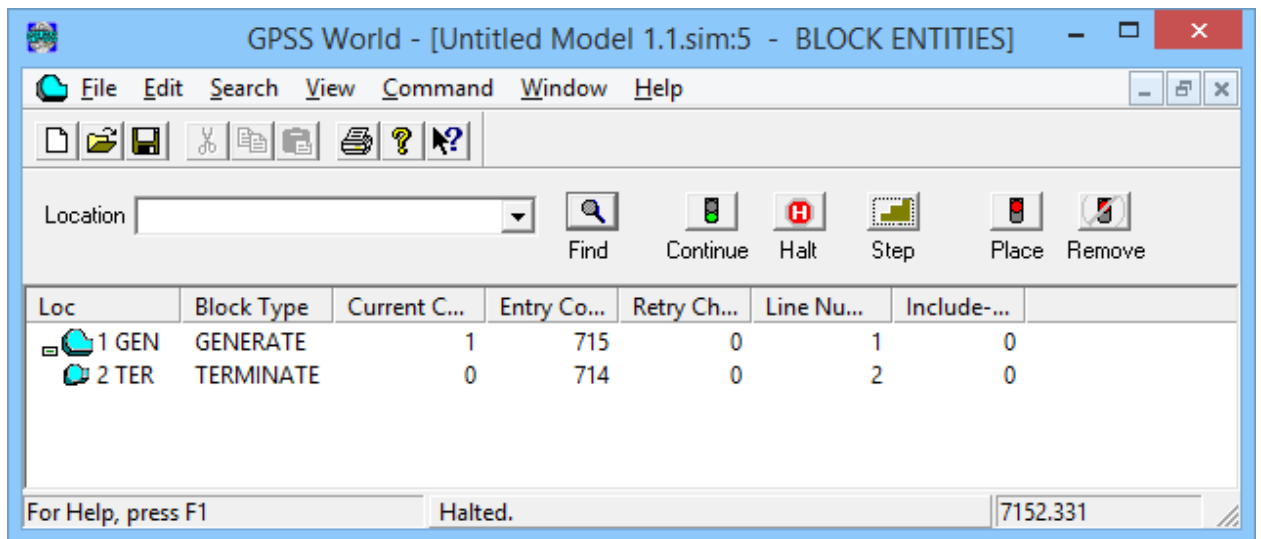
```
generate 10,3  
terminate
```

Для запуска созданной модели нужно вызвать в меню Command пункт Create Simulation или нажать Ctrl-Alt-S. Если код не содержит ошибок, то будет выведено следующее сообщение:

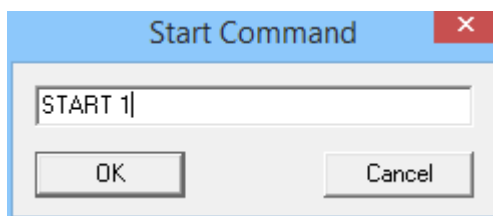
**Model Translation Begun.  
Ready.**

*\*Для наглядности рекомендую поставить галочку в View -> Simulation Clock. Это позволит проследить за временем работы системы, оно будет отображаться в правом нижнем углу.*

Симуляцию можно производить пошагово. Для этого нужно в меню Command выбрать пункт Step 1. Другой вариант, более наглядный – Window -> Simulation Window -> Blocks Window, там нажать на кнопку Step. Во втором случае можно будет проследить за перемещением транзактов по системе.



Выполнять можно и не пошагово, запустив Command -> Start. Появится окно со строкой (пока её не нужно редактировать).



Однако, если запустить текущую модель, то она будет работать бесконечно, без остановки (это можно увидеть по текущему времени). Прервать выполнение можно с помощью команды Halt.

Так вот, число после слова Start – это счетчик завершений. Если мы напишем следующий код:

```
generate 10
terminate 1
```

А затем запустим симуляцию со строкой Start 3, то произойдут следующие события:

- Симуляция начинается
- Через 10 единиц времени генерируется первый транзакт
- Первый транзакт отправляется в блок terminate; счетчик завершений уменьшается на 1 и становится равен 2
- Через 10 единиц времени генерируется второй транзакт
- Он же отправляется в terminate; счетчик становится равен 1
- Через 10 единиц времени генерируется третий транзакт
- Отправляется в terminate; счетчик становится равен 0, из-за чего выполнение симуляции прекращается.


Если бы мы запустили её со строкой Start 1, то симуляция бы завершилась через 10 единиц времени.

Собственно, такую конструкцию (generate с некоторым временем, а после него – сразу terminate 1) удобно использовать, когда у симуляции по заданию ограничено время. Например, в первой лабораторной это время ограничено рабочим днём, т.е.  $8 * 60 = 480$  минутами.

Занятие и освобождение приборов, а также задержка транзактов

Для того, чтобы смоделировать обработку детали на станке, необходимо использовать три типа блоков. Первый тип – это блок занятия устройств, Seize:

Enter Block Information

 **SEIZE**

SEIZE - Take ownership of Facility.

A:  Facility entity. Required.

B:

C:

D:

E:

F:

G:

H:

Label:


Comment:

OK Cancel Help

Единственный его параметр – это имя устройства, на котором производится задержка.

Второй – это блок задержки транзакта, Advance:

Enter Block Information

 **ADVANCE**

ADVANCE - Delay movement.

A:  New Assembly Set. Required.

B:  Time halfrange or Function.

C:

D:

E:

F:

G:

H:

Label:

Comment:

OK Cancel Help

Первые два параметра совпадают с параметрами блока Generate: среднее время задержки и разброс относительно этого среднего времени.

Третий блок – блок освобождения устройств, Release:

Enter Block Information

**RELEASE**

RELEASE - Relinquish Facility ownership.

A:  Facility entity. Required.

B:

C:

D:

E:

F:

G:

H:

Label:

Comment:

OK Cancel Help

Параметр один – имя устройства, которое нужно освободить.


Блоки Seize/Release не задерживают транзакт, но они необходимы для получения статистики по устройству, через которое проходят транзакты. Если нужно узнать только время прохождения транзакта по системе, то можно обойтись лишь блоком Advance.

#### Очереди

В случае, если в момент поступления транзакта устройство еще не освободилось, транзакт будет ожидать освобождения этого устройства. Для получения статистики по очередям используются блоки Queue и Depart.

Параметры следующие: первый – это имя очереди, второй – это количество мест, занимаемых транзактом (по умолчанию = 1).

Enter Block Information

 **QUEUE**

QUEUE - Increase the Content of a Queue Entity.

A:

B:

C:

D:

E:

F:

G:

H:

Queue entity. Required.

Increment.

Label:


Comment:

OK

Cancel

Help

Enter Block Information

 **DEPART**

DEPART - Decrease the Content of a Queue Entity.

A:

B:

C:

D:

E:

F:

G:

H:

Queue entity. Required.

Decrement.

Label:

Comment:

OK

Cancel

Help

Таким образом, если мы хотим создать следующую модель:

- Среднее время генерации транзактов = 10, разброс = 5
- Среднее время задержки на устройстве = 9, разброс = 3
- Время моделирования = 1000

То должен получиться примерно такой код (запускается со Start 1):

```
generate 10,5
queue testQueue
seize testFacility
depart testQueue
advance 9,3
release testFacility
terminate
```

```
generate 1000
terminate 1
```

В отчёте можно посмотреть статистику по устройству и очереди:

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
TESTFACILITY	100	0.907	9.072	1	101	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
TESTQUEUE	3	0	100	36	0.507	5.070	7.921	0

По устройству:

1. Facility – имя устройства
2. Entries – количество пришедших транзактов
3. Util. – процент занятости устройства
4. Ave. time – среднее время занятости устройства каждым транзактом

По очереди:

1. Queue – имя очереди
2. Max – максимальное количество элементов в очереди
3. Ave. cont. – среднее количество элементов в очереди за всё время моделирования
4. Ave. time. – среднее время пребывания транзакта в очереди

Более подробно это описано в книге Боева, в приложении №4.

Сбор статистики по другим параметрам

По заданию в первой лабораторной необходимо получить статистику по времени генерации транзактов и времени обработки. Для этого можно воспользоваться следующими блоками.

Переменные и таблицы

Переменные в GPSS определяются следующим образом:

<Имя переменной> variable <выражение>

Таблицы, в которые будут заноситься значения, определяются так:

<Имя таблицы> table V\$<имя переменной>, <начало интервала>, <шаг внутри интервала>, <количество шагов>

Внесение данных в таблицу производится блоком Tabulate. У него следующий синтаксис:

Enter Block Information


**TABULATE**

TABULATE - Collect Table Entity datum.

A:

B:

C:

D:

E:

F:

G:

H:

Table entity. Required.

Weighting factor.

Label:


Comment:

OK
Cancel
Help

tabulate <имя таблицы> <весовой коэффициент>

Для сохранения значения некоторой величины в памяти используется блок Savevalue.

Enter Block Information


**SAVEVALUE**

SAVEVALUE - Modify Savevalue Entity.

A:

B:

C:

D:

E:

F:

G:

H:

Savevalue entity (+/-). Req.

Value. Required.

Label:

Comment:

OK
Cancel
Help



Первый параметр – это имя или номер ячейки памяти, в которую будет сохраняться значение; второй – это то самое значение.

Например, для сбора статистики по времени генерации можно написать следующее:

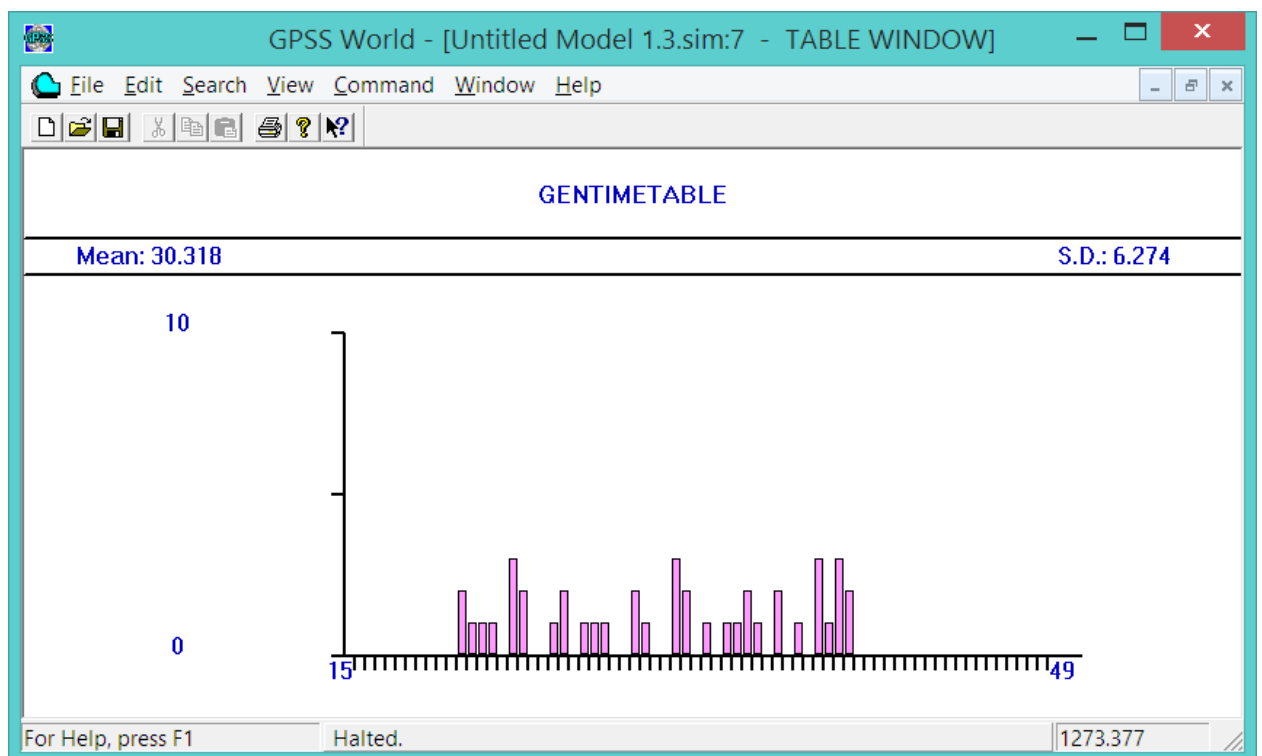
```
generationTime variable C1-X$savedTime
genTimeTable table V$generationTime,15,0.5,70
generate 30,10
tabulate genTimeTable
savevalue savedTime,C1
terminate
```

\*Пояснение – C1 – это текущее время в симуляции.

Что же происходит при запуске этого кода? Происходит следующее:

- Объявляется переменная generationTime, значение которой вычисляется как разность между текущим временем (C1) и значением, находящимся в ячейке памяти savedTime
- Объявляется таблица genTimeTable, в которую будут вноситься значения переменной generationTime, лежащие в интервале от 15 до 50 с шагом 0.5 ( $15 + 70 * 0.5 = 50$ )
- Генерируются транзакты со средним временем 30 и разбросом в 10
- В таблицу genTimeTable вносится новое значение, которое определяется как текущее значение переменной generationTime, т.е. разница между текущим временем и сохранённым в ячейке savedTime. В момент прохода первого транзакта в savedTime находится 0
- В ячейку savedTime помещается значение времени в момент прохождения этого блока, которое и будет сравниваться при проходе следующего транзакта
- Транзакт уничтожается

Содержимое таблицы можно будет посмотреть в виде гистограммы, нажав Window -> Simulation Window -> Table Window и выбрав в появившемся окне нужную таблицу.



Под заголовком таблицы находятся значения двух параметров: среднее значение (Mean) и среднеквадратическое отклонение (6,274). Высота столбиков гистограммы показывает, сколько раз транзакт попал в данный участок интервала.

Про сбор статистики по времени обработки транзактов

В случае, когда нам нужно получить статистику по каждому независимому транзакту, данные стоит сохранять не в ячейку памяти, а в параметры транзакта. Для этого используется блок Assign.

Enter Block Information

**ASSIGN**

ASSIGN - Modify XN Parameter.

A:  Parameter. Required.

B:  Value. Required.

C:  Function entity number.

D:

E:

F:

G:

H:

Label:

Comment:

OK Cancel Help

Первый параметр – имя параметра транзакта; второй – сохраняемое значение.

При работе с параметрами, а не ячейками памяти, в блоке Variable нужно использовать префикс P, а не V:

```
processingTime variable C1-P$savedTime
```

Блок Assign применяется следующим образом:

```
assign savedTime, C1
```