

Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Кафедра «Робототехнические системы»

Е.Р. Новичихина
Р.В. Новичихин

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ
В СРЕДЕ GPSS**

Методическое пособие
по дисциплине «Моделирование систем»
для студентов специальности

1-53 01 02 «Автоматизированные системы обработки информации»

Минск 2010

УДК 004.92
ББК 32.973.26-018.2
Н 73

Рецензенты:
Ю.Н. Петренко, А.Т. Кулаков

Новичихина, Е.Р.
Н 73 Имитационное моделирование систем в среде GPSS: методическое пособие по дисциплине «Моделирование систем» для студентов специальности 1-53 01 02 «Автоматизированные системы обработки информации» / Е.Р. Новичихина, Р.В. Новичихин. – Минск: БНТУ, 2010. – 42 с.

ISBN 978-985-525-386-1.

Методическое пособие предназначено для выполнения лабораторных и практических работ, связанных с имитационным моделированием систем в среде GPSS.

В пособии приводятся концепция GPSS, рекомендации по практическому моделированию, примеры программирования типовых ситуаций.

Помимо студентов пособие может быть использовано так же инженерами, желающими самостоятельно освоить язык GPSS.

УДК 004.92
ББК 32.973.26-018.2

ISBN 978-985-525-386-1

© Новичихина Е.Р.,
Новичихин Р.В., 2010
© БНТУ, 2010

ВВЕДЕНИЕ

Методическое пособие предназначено для обучения студентов основам имитационного моделирования систем на языке GPSS (General Purpose System Simulation), World Student Version 4.3.2.

Материалы пособия необходимы и достаточны для самостоятельного выполнения студентами практических и лабораторных работ по курсу «Моделирование систем». Кроме того, пособие может быть полезным при выполнении разделов курсового и дипломного проектирования, связанных с моделированием.

1 ОСНОВНЫЕ ПОЛОЖЕНИЯ GPSS

1.1 Назначение языка

GPSS является специализированным языком для имитационного моделирования дискретных процессов.

Дискретность означает, что результатом функционирования системы являются штучные продукты или услуги, а сам процесс функционирования можно представить как череду событий.

Учитываемыми и анализируемыми параметрами являются количество, время, последовательность (маршрут).

Из четырех фундаментальных функций любой системы – накопления, приемки-выдачи, передачи и преобразования – GPSS занимает вплотную первыми тремя. Элементы-преобразователи также присутствуют, но выступают в роли «черных ящиков». Языку безразлично, что именно, как и с каким качеством происходит с предметом внутри преобразователя. Ему важно только, сколько предметов, когда и при каких условиях появляется на входах-выходах элементов.

GPSS успешно заменяет все аналитические методы подобного назначения, такие как теория систем и сетей массового обслуживания, метод динамики средних, циклограммы, сети Петри.

Встроенный в GPSS язык PLUS (Programming Language Under Simulation) позволяет пользователю при необходимости создавать собственные подпрограммы, аналогично тому, как он бы это делал с использованием алгоритмических языков общего назначения. В PLUS содержатся также готовые библиотечные модули, например

генераторы случайных чисел. Кроме того, PLUS предоставляет средства автоматизации планирования имитационного эксперимента, факторного и регрессионного анализа, оптимизации.

1.2 Пример модели

Имеется станок с входным накопителем емкостью на 6 деталей (рис. 1). Детали поступают на обработку через случайные промежутки времени $T_{\text{пост}} = 18 \pm 6$ мин. Время обработки также случайное $T_{\text{обр}} = 20 \pm 4$ мин. Требуется определить показатели функционирования этой обрабатывающей системы после обработки 1000 деталей.

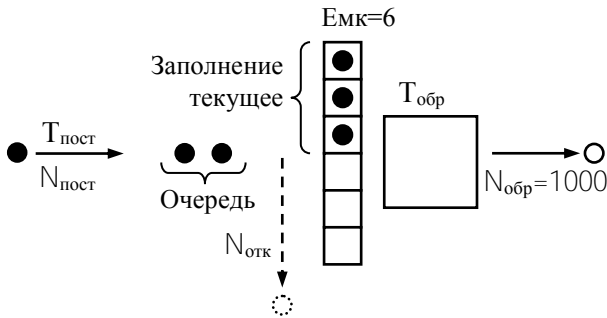


Рис. 1. Схема объекта моделирования

Текст программы GPSS будет выглядеть следующим образом (после символа «;» – комментарии):

Накоп	STORAGE	6	; Задать имя и емкость накопителя
	GENERATE	18, 6	; Генерировать поступление деталей
	QUEUE	Otcered	; Поставить деталь в очередь
	ENTER	Nakop, 1	; Занять деталью 1 место в накопителе
	DEPART	Otcered	; Удалить деталь из очереди
	SEIZE	Stanok	; Перевести деталь на станок
	LEAVE	Nakop, 1	; Освободить 1 место в накопителе
	ADVANCE	20, 4	; Задержать (обработать) деталь
	RELEASE	Stanok	; Освободить станок
	TERMINATE	1	; Вывести 1 деталь из модели
	START	1000	; Начать моделирование, завершить ; после выхода 1000 деталей

После прогона модели будет выдан стандартный отчет (REPORT) с результатами (рис. 2).

Обратим внимание, что при написании программы нам не потребовалось специально разрабатывать алгоритм, мы просто провели деталь по ее маршруту в «живой» системе. Все остальное сделает GPSS. Он сам будет осуществлять нужные поступления, продвижения, размещения, задержки и удаления деталей. Сам будет отслеживать занятость станка и заполнение накопителя. Сам будет собирать статистику по выпуску деталей, по их очередям, по продолжительности нахождения их в системе и пр. В программе нам даже не надо указывать требуемые выходные данные. Все возможные первичные результаты имитации GPSS выдаст автоматически. При необходимости, в стандартный отчет пользователь может включить и производные результаты, задав их в тексте программы формулами.

1.3 Объекты языка

Объекты GPSS (Entities) – это элементы, из которых язык строит модели. Их перечень приведен в табл. 1.

Динамические объекты представлены единственным типом – *транзактами* (ТА). ТА имитируют предметы деятельности системы, продвигаются по ее структурным элементам и инициируют различные события в ней. В нашем примере ТА – это детали. ТА-тами могут быть также клиенты, работы, задачи, заказы, документы и пр. Аналог ТА – понятие «заявка» в теории массового обслуживания.

Операционные объекты – это блоки и команды. *Блоки* описывают структуру и действия моделируемой системы, а *команды* управляют самой моделью. В GPSS около 50 блоков и 25 команд. Операционные объекты вбирают в себя все остальные представленные в табл. 1 объекты, кроме ТА.

Аппаратные объекты – это операционные объекты, описывающие структурные элементы системы. Они бывают трех типов: устройства, многоканальные устройства (МУ) и логические ключи. В нашем примере устройством является станок, а МУ – накопитель.

GPSS World Simulation Report - Untitled Model 1.1.1
 Friday, December 16, 2005 13:46:54

START TIME **END TIME** BLOCKS FACILITIES STORAGES
 0.000 20093.623 9 1 1

NAME	VALUE	
NAKOP	10000.000	
OTCERED	10001.000	Число
STANOK	прошедш. 10002.000	

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1121	0	0
	2	QUEUE	1121	115	0
	3	ENTER	1006	0	0
	4	DEPART	1006	5	0
	5	SEIZE	1001	1	0
	6	LEAVE	1000	0	0
	7	ADVANCE	1000	0	0
	8	RELEASE	1000	0	0
	9	TERMINATE	1000	0	0

Приборы	Число обл.	Ки	Ср.вр.обсл.	Готовн.	№ТА на обл.
FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAIL.	OWNER
STANOK	1001	0.999	20.051	1	1001

Очереди	Макс. Тек.	Ч.занят.	Не ждали	Ср.очер.	Ср.вр. в оч.	
QUEUE	MAX	CONT.	ENTRY	AVE.CONT.	AVE.TIME	
OTCERED	115	115	1121	47	54.362	974.417

Накопители	Емк.	Тек.	Мин.	Макс.	Ч.занят.	Гот.	Ср.содерж.	Ки
STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
NAKOP	6	0	0	6	1006	1	5.882	0.980

Цепь тек. событий:

№ТА	№приор.	Вр.входа	№ансамб.	№тек.бл.	№сл.бл.	№пар.	Знач.пар.
SEC XN	PRI	M1	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
1001	0	7915.915	1001	5	6		

Цепь буд. событий:

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
1122	0	20101.639	1122	0	1		

Рис. 2. Пример стандартного отчета GPSS

Таблица 1

Объекты языка

Категория	Тип	В английском оригинале	Синонимы
Динамическая	Транзакты (ТА)	Transaction	Сообщения, заявки
Операционная	Блоки	Block	Элементы и действия
	Команды	Commands	Управление
Аппаратная	Устройства	Facility	Приборы
	Многоканальные устройства (МУ)	Storage	Память, накопители
	Логические ключи	Logicswitch	Переключатели
Вычислительная	Переменные (целые, действительные, логические)	Variable, Fvariable, Bvariable	
	Функции	Function	
	Выражения	Expression	
	Процедуры, подпрограммы	Procedure	
	Эксперимент	Experiment	
Статистическая	Очереди	Queue	
	Таблицы (для гистограмм)	Table	
	Стандартные числовые атрибуты (СЧА)	System Numerical Attributes (SNA)	Системные параметры
Запоминающая	Сохраняемые величины	Savevalue	Ячейки
	Матрицы величин	Matrix	Массивы
Группирующая	Цепи пользователя	Userchain	Списки
	Группы	Numeric Group, Transaction Group	

Если бы работало несколько параллельных станков, их тоже можно было бы представить в виде одного МУ. Модельное отличие МУ от нескольких параллельно работающих устройств заключается в том, что в МУ каналы обезличены и ТА не может выбрать конкретный, а должен следовать на любой из свободных. Кроме того, канал МУ не может быть захвачен ТА-ом с прерыванием обслуживания текущего ТА.

1.4 Формализация объекта моделирования

Каждому предмету функционирования системы ставится в соответствие *транзакт* языка. За каждым ТА закреплен определенный состав параметров, которые называются *атрибутами*. Предметы отличающихся типов и состояний описываются в GPSS ТА-ами с разными значениями атрибутов.

Каждому структурному элементу и действию моделируемой системы ставится в соответствие подходящий *блок* языка. Каждый блок также имеет свои атрибуты, состав которых фиксирован. Настройка блока на реальный объект осуществляется заданием конкретных значений постоянным атрибутам. Переменные атрибуты отражают изменение состояния блоков уже в процессе имитации. Большинство этих изменений осуществляется автоматически.

Для облегчения формализации структуру системы и ее действия можно (но необязательно) описать *блок-схемой* GPSS. На рис. 3 изображена блок-схема для нашего примера. Каждый блок обозначает определенный элемент или действие и имеет свое условное изображение. Эти изображения приведены ниже в разделе «Блоки и команды GPSS».

Схема напоминает обычную блок-схему алгоритма. Она повторяет путь предмета-ТА по структурным элементам-блокам в «живой» системе. Если на определенном этапе продвижения предмета в системе нужно выполнить какое-нибудь нематериальное действие (вычислить, проверить условие, принять решение по альтернативному маршруту и пр.), то эти действия условно материализуются также соответствующими блоками. В блок надо фиктивно зайти и тогда это действие будет выполнено. При этом в системе предмет физически никуда не заходит. Последовательность проводки ТА по блокам задается линиями потока, которые изображаются стрелками.

По готовой блок-схеме код программы пишется машинально. Каждый блок просто заменяется одноименным оператором. Так как названия операторов в переводе очень точно передают их смысл, то кодирование приближается к написанию текста на естественном языке и необходимость в блок-схеме часто отпадает.

Функционирование системы представляется в GPSS как череда событий. *Событием* называется изменение состояния (атрибутов) любого из объектов модели.

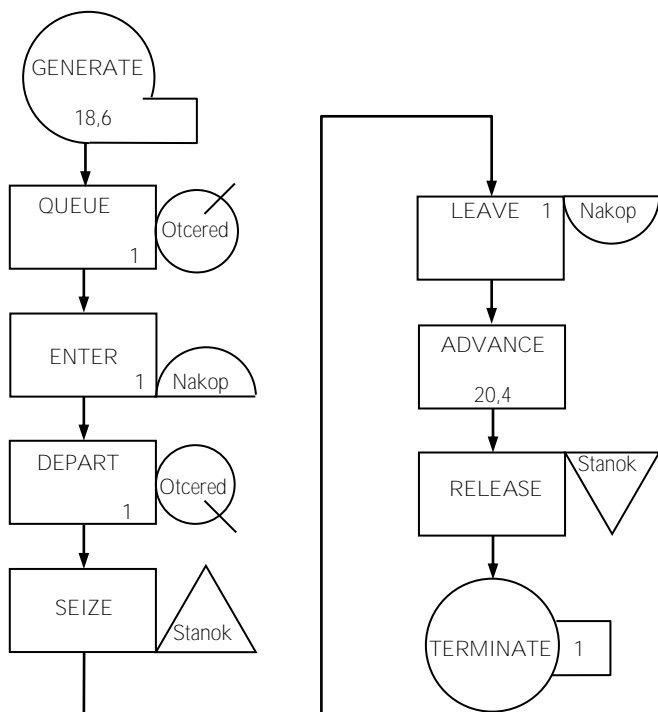


Рис. 3. Пример блок-схемы модели GPSS

Реальное время моделируется в GPSS *системными часами*. Язык работает с условными единицами времени. Их интерпретацию осуществляет сам пользователь (час, минута, секунда, микросекунда). Какие единицы он имел в виду в исходных данных, в таких же единицах нужно трактовать временные результаты.

1.5. Принцип построения программы

Текст программы получается кодированием блоков схемы одноименными *операторами*. Каждому оператору отводится отдельная строка. В дополнение к схеме, в текст программы включаются строки-операторы необходимых описателей и команд.

Последовательность операторов должна совпадать с маршрутом ТА. Для неоднозначных маршрутов или нестандартных ситуаций предусматриваются операторы принудительных переходов ТА, отличных от основного потока.

В одной модели могут учитываться ТА и обслуживания разного вида. Фрагмент блок-схемы или текста программы, относящийся к ТА и обслуживанию одного вида, называется *сегментом*. Обычно сегмент заключен между блоками-операторами GENERATE и TERMINATE.

Последовательность блоков-операторов внутри сегмента имеет принципиальное значение. Последовательность же самих сегментов в программе безразлична. GPSS будет их отрабатывать во времени параллельно, от текущего события к ближайшему без пропусков, к какому бы сегменту это событие не относилось.

Один и тот же аппаратный блок или очередь может присутствовать в нескольких сегментах.

ТА могут принудительно передаваться из одного сегмента в другой. ТА так же могут синхронизироваться, блокироваться или объединяться с ТА других сегментов.

1.6 Принцип имитации

Имитация работы системы осуществляется специальной встроенной программой GPSS, которая называется *симулятором*.

Симулятор автоматически, без детального программирования со стороны пользователя осуществляет:

- генерацию ТА-ов по нужному временному закону, ввод их в модель и вывод после отработки;
- продвижение ТА от блока к блоку в соответствии с маршрутом и логикой;
- задержку ТА в блоках на требуемое время, до освобождения следующего блока или до выполнения определенных условий;
- фиксацию или инициирование событий, сопутствующих продвижению ТА;
- изменение значений зависимых атрибутов;
- сбор по ходу моделирования всевозможной статистики и выдачу в конце стандартного отчета с результатами.

Образно принцип работы GPSS можно представить, как изображено на рис. 4.

Для отслеживания и осуществления изменений в системе программа-симулятор использует атрибуты. *Атрибуты* – это фиксированный для данного объекта GPSS состав параметров. Атрибуты можно сравнить со стандартными формулярами, которые закрепляются на все время моделирования за каждым из объектов модели, включая каждый отдельный ТА.

После генерации или очередной активизации ТА, программа-симулятор продвигает его из предыдущего блока в следующий в соответствии с программой пользователя. Так по рис. 4 активный на текущий момент ТА №3 поступит на очередной по тексту программы блок. Это может быть очередь, прибор, многоканальное устройство. Есть блоки, в которые ТА заходит условно, из изображенных на рисунке – это блок «логический ключ». В этот блок ТА заходит, чтобы изменить его состояние – включить, выключить или инвертировать. Могут быть и другие блоки с условным заходом. Например, в блок «сохраняемая величина» ТА заходит, чтобы изменить или пересчитать ее значение. Также условно ТА может заходить в специальные блоки, чтобы изменить собственные атрибуты. По программе ТА условно заходит и в операторы-команды.

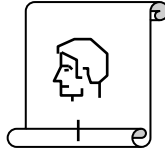
В своей работе симулятор придерживается ряда принятых в GPSS соглашений:

- каждый блок имеет определенное количество входов и выходов;
- ТА может находиться только внутри блока;
- действие, описываемое соответствующим блоком, выполняется при входе ТА в него;
- блок может отказать ТА во входе при каких-то блокирующих условиях, в этом случае ТА будут накапливаться в предыдущем блоке до снятия блокировки или направляться в другой специально указанный блок;
- движения (активизация) ТА-ов происходят только по одному;
- активный ТА может находиться только в одном блоке;
- изменение атрибутов объекта может происходить только в момент входа активного ТА в данный или в другой, логически связанный с ним блок.

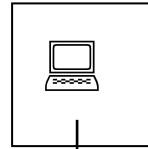
СИСТЕМА



Программа пользователя



Программа-симулятор



МОДЕЛЬ

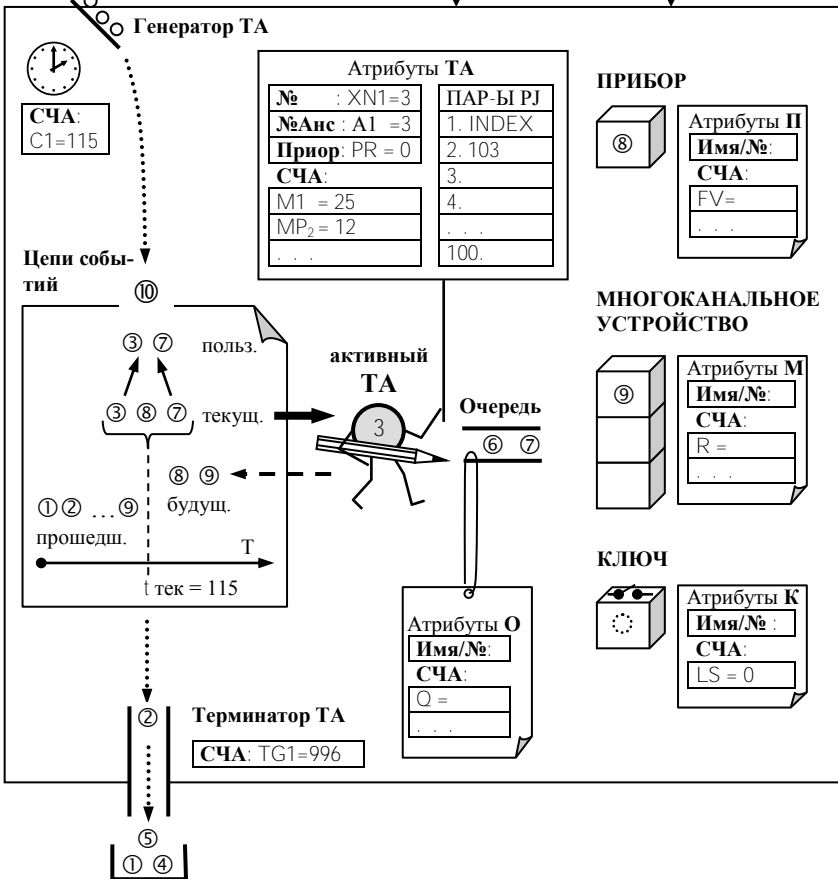


Рис. 4. Концепция GPSS и атрибуты его объектов

Последнее положение образно можно представить на рис. 4 так, как будто единственный карандаш, которым можно записывать или переписывать атрибуты, находится у активного ТА, а тот в свою очередь всегда в единственном числе. Активный ТА может переписывать и собственные атрибуты, но опять-таки, только при входе в соответствующий блок.

1.7 Цепи событий

Для упорядочивания работы симулятора все ТА помещаются в динамические списки, которые называются цепями. Виды цепей:

- цепь текущих событий;
- цепь будущих событий;
- цепь прерываний;
- цепь парных ТА;
- цепь пользователя.

Цепь текущих событий содержит ТА-ы с временем события, равным модельному времени. Дисциплина обработки ТА-ов в цепи – по приоритету. Соответственно ТА с большим номером приоритета будут автоматически располагаться в начале списка. Внутри класса ТА с одинаковым приоритетом они располагаются согласно времени поступления (первый пришел – первый обработан). Приоритет присваивается каждой ТА при ее генерации и сопровождает ее до выхода из модели. В процессе моделирования приоритет может быть программно изменен.

Цепь будущих событий содержит ТА, время движения которых больше текущего модельного времени, т.е. они должны произойти позже. Список – в порядке времени наступления. Из цепи будущих событий ТА переходят в цепь текущих событий, когда наступает их время.

Цепь прерываний содержит ТА, обработка которых была прервана. Данные ТА вводятся в модель после снятия прерывания.

Цепь парных ТА-ов содержит ТА, дальнейшее движение которых невозможно из-за отсутствия одного или нескольких других ТА.

Цепь пользователя содержит ТА, извлекаемые из модели программно самим пользователем. Эта цепь служит для моделирования различных дисциплин обслуживания.

1.8 Атрибуты объектов

Как уже говорилось, исходные свойства объектов и их состояния в данный момент времени описываются атрибутами. Состав атрибутов для объектов данного класса фиксирован.

Значения некоторых из атрибутов (P, PR, FN) задаются пользователем до моделирования. Это исходные данные, заключенные в операндах операторов языка.

Значения других атрибутов (BV, V, X, MX, а также исходные P, PR) могут задаваться и изменяются пользователем уже в процессе моделирования программно.

Значения же большинства атрибутов присваиваются, фиксируются или вычисляются симулятором автоматически. Те из них, к которым пользователь имеет доступ, называются *стандартными числовыми атрибутами (СЧА)* (System Numerical Attributes –SNA). Все СЧА доступны, однако, не все из них можно принудительно изменять.

Значение СЧА пользователь может применять для расчетов, проверок, адресации. Обращение к конкретному СЧА осуществляется по стандартному уникальному символьному обозначению. Список этих обозначений приведен ниже в разделе «Синтаксис языка».

При обращении к СЧА из программы, возвращается его значение. Например, обратившись к атрибуту XN1, мы получим номер активного в данный момент ТА, к М1 – время его пребывания в модели и т.д.

Особое место среди СЧА занимают СЧА ТА-ов, как инициаторов событий, а среди СЧА ТА-ов – группа, которая называется параметрами. *Параметры* – это зарезервированные за каждым ТА 100 «пустых» ячеек для нужд пользователя. Смысл и обозначение параметров языком не регламентируется. Для обращения к параметру используется только его порядковый номер с признаком (P либо *) того, что номер относится именно к параметрам ТА, например, P12 либо *12. Что именно и в какую ячейку класть, решает сам пользователь. Например, в параметр №2 ТА-ов мы запишем (при помощи оператора ASSIGN после GENERATE) имя/номер прибора для альтернативного маршрута в случае нештатной ситуации. В тексте программы закладываем проверку возникновения такой ситуации (оператором TEST или GATE) и указываем, что в случае выполне-

ния условия адрес для последующего продвижения ТА нужно взять из P2. В противном случае все должно идти по плану, т.е. к следующей строке-блоку.

Параметры введены в GPSS для повышения гибкости языка и в программах простых моделей могут не понадобиться.

1.9 Ансамбли транзактов

Ансамбль – это группа чем-то связанных между собой ТА. Все члены одного ансамбля так и называются – *связанные ТА* (related Transactions). Аналог ансамбля – семья.

Ансамбли нужны, что бы выделять ТА для выборочных действий. Например, собирать изделие только из определенных деталей, а не из любых пришедших.

Операции объединения нескольких ТА в одну (оператор ASSEMBLE), выпуска после накапливания (оператор GATHER) или синхронно с другими (оператор MATCH) возможны только с ТА одного ансамбля.

Принадлежность данного ТА к тому или иному ансамблю определяется/задается СЧА **A1**, значение которого и является номером ансамбля (Assembly Set).

Значение **A1** присваивается каждой ТА по умолчанию при генерации (оператором GENERATE) и принимается равным значению уникального номера самой ТА ($A1 = XN1 = 1,2,\dots$).

Т.о. любой ТА уже сам по себе является ансамблем минимального единичного состава. Таковым он может и остаться. Его **A1** при этом сохранится.

ТА может расширить свой ансамбль порождением точно таких же *ТА-потомков* (оператором SPLIT). Их **A1** при этом примет значение как у *ТА-родителя*.

ТА может принять в свой ансамбль другие ТА (оператором ADOPT). Их **A1** при этом примет значение как у *ТА-усыновителя*.

ТА может сам влиться в другой ансамбль (оператором ADOPT). Его **A1** при этом примет значение номера этого ансамбля, т.е. изменится, но XN1 – естественно сохранится.

ТА может, наконец, образовать с другими ТА новый ансамбль (оператором ADOPT). Значения **A1** всех их при этом изменится на указанное пользователем.

2 СИНТАКСИС ЯЗЫКА GPSS

2.1 Алфавит

Для написания программ на GPSS используются латинские буквы [A...Z] и цифры [0...9], а также специальные символы. Кириллицу допустимо использовать только в комментариях (после символов «*» или «;»).

2.2 Числа

GPSS использует, как правило, целые числа. При необходимости можно использовать и действительные числа. Их признак – наличие десятичной точки.

2.3 Формат строки программы

Состав, последовательность и размер элементов строки приведены в табл. 2.

Таблица 2

Формат строки программы

Поля (столбцы)					
*	Метки	Операторы (блоки и команды)	Операнды (до 8 шт)	;	Комментарии
Line number	Label	Verb	Operands		Comment
Позиции (колонки)					
1	2...6	8...18	19...73	74	75...

В рассматриваемой версии языка приведенное число позиций (форматирование) соблюдать необязательно, главное, чтобы между полями присутствовал хотя бы один пробел или табуляция.

Некоторые поля могут отсутствовать. Строка минимального состава может состоять только из оператора. Последовательность присутствующих полей должна соблюдаться.

Символ «*» используется, если вся строка является комментарием.

Метка используется для задания имени блока, для идентификации объекта или указания места в программе для перехода. Метка минимальной длины состоит из трех буквенных символов. Далее могут следовать в любой последовательности буквенные и цифровые символы, а также специальный символ разделитель «_» (но не тире и не пробел).

Оператор – это название блока или команды, условленное соглашением языка. Оператор состоит из одного, иногда – двух зарезервированных слов.

Операнд – это числовые, логические или строковые характеристики оператора, его конкретизация.

Операнд может содержать до 8 позиций, которые обозначаются буквами: A,B,C,D,E,F,G,H. Что именно подставлять в каждую позицию вместо этих букв, подскажет встроенный справочник языка (Help).

Операнды разделяются между собой запятыми. Запятые остаются и в случае, если какой-нибудь операнд отсутствует (не инициализируется или принимается по умолчанию). При этом запятые следуют без пробела. После последнего из заданных операндов запятые допускается опускать.

Символ «;» используется, если комментарий следует сразу за операндом в той же строке.

2.4 Операторы

Арифметические операторы (действия):

+	– плюс	
–	– минус	
/	– деление	
#	– умножение	(а не «*» !)
^	– возведение в степень	
\	– деление целочисленное	(10 \ 7 = 1)
@	– деление по модулю (остаток)	(10 @ 7 = 3)
)	– скобки	
.	– разделитель десятичной дроби	(а не «.» !)

Условные операторы (отношения):

>	или	'G'	– больше	(Greater)
<		'L'	– меньше	(Less)
=		'E'	– равно	(Equal)
!=		'NE'	– не равно	(Not Equal)
>=		'GE'	– больше или равно	(Greater or Equal)
<=		'LE'	– меньше или равно	(Less or Equal)

Булевы операторы:

	либо	'OR'	– логическое сложение	(ИЛИ)
&		'AND'	– логическое умножение	(И)

Очередность выполнения операторов:

^ ; # / \ ; @ ; + - ; >= <= > < ; = = ; & ; | .

Логические операторы (состояние объектов) приведены в табл. 3.

Таблица 3

Логические операторы

Объекты	Условия		Значения
Приборы	F	Используется?	1 – занят, 0 – свободен
	FI	В состоянии прерывания?	1 – обслуживает прерывание 0 – нет
Накопители	SF	Заполнен?	1 – заполнен полностью 0 – есть хотя бы один свободный канал
	SE	Пуст?	1 – пуст 0 – есть хотя бы один занятый канал
Логические ключи	LS	Установлен?	1 – установлен 0 – сброшен

2.5 Стандартные числовые атрибуты (СЧА)

Обозначения СЧА приведены в табл. 4. Здесь фрагмент <...> обозначает *идентификатор*, т. е. имя числовое или символическое, метку или № параметра ТА, где записано имя требуемого блока.

Таблица 4
Стандартные числовые атрибуты (СЧА) GPSS

Объект	СЧА	Значение	Что означает
1	2	3	4
ТА (активн.)	XN1	Целое	№ активного ТА
	Pj или *j	Целое, действ., строковое	Значение j-го параметра активного ТА
	PR	Целое [0,127]	Значение приоритета
	M1	Действ.	Время пребывания ТА в модели [C1–Генерац]
	MPj	Действ.	Время с момента входа в блок [C1–Pj] j – № параметра ТА, где зафиксировано Tvx
	A1	Целое	Номер ансамбля активного ТА
Блоки	MB<...>	0 либо 1	Состояние синхронизации (1- в блоке есть ТА из того же ансамбля, что и активный)
	N <...>	Целое	Счетчик вхождений (общее кол. ТА)
Приборы	W <...>	Целое	Текущее содержимое (кол. ТА)
	F <...>	0 либо 1	Состояние прибора (1– занят)
	FV <...>	0 либо 1	Состояние прибора (1– доступен)
	FI <...>	0 либо 1	Состояние прибора (1– в прерывании)
	FC<...>	Целое	Счетчик числа занятый
	FR <...>	Действит. [0,1000]	Коэффициент использования прибора (в долях от 1000)
МУ	FT<...>	Действит.	Среднее время занятия прибора одним ТА-ом
	R <...>	Целое	Количество свободных каналов
	S <...>	Целое	Текущее содержимое
	SC<...>	Целое	Счетчик общего числа вхождений

Продолжение таблицы 4

1	2	3	4
	SE<...>	0 либо 1	Состояние МУ (1– пуст весь)
	SF<...>	0 либо 1	Состояние МУ (1– заполнен полностью)
	SV<...>	0 либо 1	Состояние МУ (1– доступен)
	SM<...>	Целое	Максимальное содержимое
	SA<...>	Действит.	Средневзвешенное содержимое
	SR<...>	Действит. [0,1000]	Коэффициент использования МУ (в долях от 1000)
	ST<...>	Действит.	Среднее время нахождения ТА-а в МУ
Логич. ключи	LS<...>	0 либо 1	Состояние (0– сброшен 1– установлен)
Переменные	BV<...>	0 либо 1	Значение булевой переменной
	V <...>	Целое, действит.	Значение арифметической переменной
Функции	FN<...>	Действ.	Значение функции
Очереди	Q <...>	Целое	Текущее содержимое очереди
	QM<...>	Целое	Максимальное содержимое
	QA<...>	Действит.	Среднее содержимое
	QC<...>	Целое	Счетчик общего числа вхождений
	QZ<...>	Целое	Счетчик нулевых вхождений (не ждали)
	QT<...>	Действит.	Среднее время пребывания по QC
	QX<...>	Действит.	Среднее время пребывания по QZ
Таблицы	TB<...>	Действит.	Среднее значение аргумента таблицы
	TC<...>	Целое	Входной счетчик
	TD<...>	Действит.	Среднеквадратическое отклонение аргумента
Сохран. величины	X <...>	Целое, действит., строковое	Значение сохраняемой величины
Матрицы величин	MX<...> (m,n)	Целое, действит., строковое	Значение элемента m-ой строки n-ого столбца матрицы
Цепи пользо- вателя	CA<...>	Действит.	Среднее содержимое
	CC<...>	Целое	Счетчик общего числа вхождений в цепь
	CH<...>	Целое	Текущее содержимое
	CM<...>	Целое	Максимальное содержимое
	CT<...>	Действит.	Среднее время пребывания в цепи

1	2	3	4
Группы	GN<...>	Целое	Номер членства числовой группы
	GT<...>	Целое	Номер членства группы ТА-ов
Генера- торы	RNi	Действит., [0,1[, Целое [0,999]	В качестве аргумента функции В других случаях, i – № генератора
Время	AC1	Действит.	Абсолютное системное время (после CLEAR)
	C1	Действит.	Относительное системное время (после RESET)
Общес- темные	TG1	Целое	Счетчик оставшихся завершений
	Z1	Целое	Свободная память

2.6 Методы адресации

В GPSS используется два типа адресации – прямая и косвенная.

Прямая адресация – это обращение к конкретному объекту или величине по его идентификатору. Идентификатором может служить символическое имя, числовое имя (порядковый номер, уникальный внутри своего типа объектов) или метка.

Косвенная адресация – это обращение не к самому объекту напрямую, а к № параметра ТА, где «лежит» идентификатор требуемого объекта. Значение параметра может меняться, что позволяет осуществлять вариантность маршрутов и величин.

Примеры адресации представлены в табл. 5.

Обратим внимание, что в таблице 5 при всех вариантах адресации речь идет об одном и том же 1-ом приборе «Stanok», об одном и том же 5-ом МУ «Sklad», об одном и том же значении «4» арифметической переменной. То, что имеется в виду именно прибор под №1 и МУ под №5, а не скажем очереди под эти же номерами, языку понятно, т.к. для блока «SEIZE» и для СЧА «R» другие варианты исключены.

Примеры адресации

Прямая		Косвенная по параметру ТА	Комментарии
по символическому имени	по числ. имени		
SEIZE Stanok	SEIZE 1	SEIZE *6 или SEIZE P6	Занять прибор по имени «Stanok» Занять прибор под №1 Занять прибор, № которого указан в параметре №6 ТА
R\$Skład	R5	R*2 или RP2	Оставшаяся емкость МУ по имени «Skład» Оставшаяся емкость МУ под №5 Оставшаяся емкость МУ, № которого указан в параметре №2 ТА
Kol VARIABLE 4 V\$Kol	V17	V*3 или VP3	Значение переменной под именем «Kol» Значение переменной под №17 Значение переменной, № которой указан в параметре №3 ТА

3 БЛОКИ И КОМАНДЫ GPSS

В GPSS около 50 блоков и 25 команд. Наиболее часто встречающиеся из них, приведены в табл. 6.

Форматы блоков и команд приведены в табл. 7.

Условные графические обозначения блоков для блок-схем приведены в табл. 8.

Таблица 6

Основные блоки и команды GPSS

К	Тип	Оператор	Перевод	Назначение
1	2	3	4	5
А	Приборы	SEIZE	ЗАНЯТЬ	Перевести ТА на прибор после его освобождения
		ADVANCE	ЗАДЕРЖАТЬ	Задержать ТА на определенное время
		RELEASE	ОСВОБОДИТЬ	Освободить прибор (после SEIZE)
		PREEMPT	ЗАХВАТИТЬ	Захватить прибор ТА-ом с большим приоритетом, с прерыванием обрабатываемого ТА
		RETURN	ВЕРНУТЬ	Возвратить прибор прерванному ТА (после PREEMPT)
МУ	STORAGE*	ЕМКОСТЬ	Задать емкость МУ (число каналов)	
	ENTER	ВОЙТИ	Ввести ТА в МУ и занять определенное число мест емкости	
	LEAVE	ВЫЙТИ	Вывести ТА из МУ и освободить определенное число мест	
Логич. ключи	LOGIC	ЛОГИКА	Задать значение логического переключателя (0, 1, инверсия)	
Блок любой	EXECUTE CHANGE	ИСПОЛНИТЬ ПОМЕНЯТЬ	Исполнить Поменять	
Д	Создание, уничтожение	GENERATE	ГЕНЕРИРОВАТЬ	Создать ТА и запустить его в модель
		TERMINATE	ЗАВЕРШИТЬ	Уничтожить ТА и вывести его из модели. Уменьшить значение счетчика завершенный
		START*	СТАРТОВАТЬ	Начать моделирование и завершить по кол. ТА на выходе
		SPLIT	РАСЩЕПИТЬ	Породить от одного ТА несколько ТА-копий и рассматривать их <i>по отдельности</i>
		ASSEMBLE	СОЕДИНИТЬ	Накопить несколько ТА ансамбля, слить их и выпустить <i>как один</i> (отличать от MATCH и GATHER)
	Ансам.	ADOPT	ПРИНЯТЬ	Изменить № ансамбля акт.ТА
Здержка	ADVANCE	ЗАДЕРЖАТЬ	Задержать ТА на определенное время (необязательно на приборе)	

Продолжение таблицы 6

1	2	3	4	5
Д	Изменение маршрута	TRANSFER	ПЕРЕДАТЬ	Переход на блок обязательный: безусловный – на единственную метку, долевой или с предпочтением – на одну из двух меток, с предпочтением по списку – на одну из нескольких меток (отличать от TEST и GATE)
		LOOP	ЦИКЛ	Повторные входы в один блок того же ТА (использовать совместно с ASSIGN)
		TEST	ПРОВЕРИТЬ	Переход на один из двух блоков по арифметическому условию (>, ≥, =, ≠, ≤, <)
		GATE	ВПУСТИТЬ	Переход на один из двух блоков по логическому состоянию ключа, прибора, накопителя, ТА
		SELECT	ВЫБРАТЬ	Выбрать один элемент из нескольких по условию, запомнить его № в параметре ТА
	Синхронизация	MATCH	СИНХРОНИЗИРОВАТЬ	Ожидать связанный ТА ансамбля на другом (сопряженном) блоке, после чего выпустить их одновременно (но каждый своим путем)
		GATHER	СОБРАТЬ	Накапливать ТА-ы одного ансамбля в этом блоке до определенного количества, после чего выпустить всю группу одновременно
	Группиров.	LINK	СВЯЗАТЬ	Ввести ТА в цепь пользователя
		UNLINK	РАСЦЕПИТЬ	Вывести ТА из цепи пользователя
	Изменение атрибутов	ASSIGN	НАЗНАЧИТЬ	Задать/изменить значение параметра ТА с указанным №
		INDEX	ИНДЕКС	Тоже, для параметра №1 ТА (= заданный параметр + число)
		PRIORITY	ПРИОРИТЕТ	Задать/изменить приоритет ТА (0...127)
MARK		ЗАСЕЧЬ	Засечь время входа ТА в блок в параметре ТА	

Окончание таблицы 6

1	2	3	4	5
С	Оче- реди	QUEUE DEPART	ВСТАТЬ ПОКИНУТЬ	Поставить ТА в очередь, заняв заданное число мест Удалить ТА из очереди, освободив заданное число мест
	Сохраняемые вели- чины	INITIAL*	НАЧАЛЬНОЕ	Установить начальное значение сохраняемой величины, ключа или матрицы
		SAVEVALUE	СОХРАНИТЬ	Задать значение сохраняемой величины (будет выведено в результирующем REPORT)
		MATRIX*	МАТРИЦА	Описать массив сохраняемых величин
MSAVE- VALUE	СОХРАНИТЬ В МАССИВЕ	Задать значение элементу массива (аналогично SAVEVALUE)		
Таб- лицы	TABLE*	ТАБЛИЦА	Описать таблицу для гистограммы указанной переменной	
	TABULATE	ТАБУЛИ- РОВАТЬ	Зафиксировать факт попадания значения переменной в соответствующий диапазон гистограммы	
В	Пере- мен- ные	VARIABLE*	ЦЕЛАЯ	Задать/изменить/рассчитать значение целой переменной
		FVARIABLE* BVARIABLE* EQU*	ДЕЙСТВ. БУЛЕВА ПОЛЬЗО- ВАТЕЛЯ	Тоже, действительной Тоже, булевой Задать значение переменной пользователя (обычно для варьируемых исходных данных)
	Функ- ции	FUNCTION*	ФУНКЦИЯ	Описать функцию парами координат X,Y для графика интерполяции (используется вместо формулы в VARIABLE, GENERATE)
П	Проце- дуры	PLUS	Язык програм- мирования при моделир.	Программирование процедурное. Библиотечные модули. Организация и обработка эксперимента

Примечания: К – категория (А – аппаратная, Д – динамическая, С – статистическая, В – вычислительная, П – процедурная); * – команды (остальное – блоки).

Таблица 7

Формат основных блоков и команд GPSS

Тип	Оператор	Структура записи и операнды
1	2	3
Прб.	SEIZE	SEIZE <Имя прибора>
	ADVANCE	ADVANCE Тср,ΔТ
	RELEASE	RELEASE <Имя прибора>
	PREEMPT	PREEMPT <Имя прибора>,{_/_PR},<Имя блока для прерванного ТА>
	RETURN	RETURN <Имя прибора>
МУ	STORAGE*	<Имя МУ> STORAGE Емкость
	ENTER	ENTER <Имя МУ>, Количество занимаемых мест
	LEAVE	LEAVE <Имя МУ>, Количество освобождаемых мест
Кл.	LOGIC	LOGIC <Имя ключа> {R/S/I}
ТА	GENERATE	GENERATE Тср,ΔТ,Тпервого,Количество ТА,№ приоритета
	TERMINATE	TERMINATE Количество мест
	START*	START Количество завершений (выходовТА)
	SPLIT	SPLIT Количество потомков,<Имя блока для направления потомков>
	ASSEMBLE	ASSEMBLE Количество ТА
	ADOPT	ADOPT № ансамбля
	ADVANCE	ADVANCE Тср,ΔТ
	TRANSFER	TRANSFER {_/_Вероятн2/BOTH/ALL},<Имя бл.1>,<Имя бл.2>,Шаг
	LOOP	LOOP № параметра ТА, где записано количество повторов,<Имя бл.возврата>
	TEST	TEST {G/GE/E/NE/LE/L} <Имя ЧА1>,<Имя ЧА2>,<Имя блока если условие не выполняется>
	GATE	GATE {LR/LS/U/NU...} <Имя аппарата>,<Имя бл. если усл. не выпол.>
	SELECT	SELECT {G/GE/E/NE/LE/L} № параметра ТА,MIN,MAX,....
MATCH	MATCH <Имя сопряженного блока>	
GATHER	GATHER Количество ТА	
ТА	LINK	LINK <Имя цепи>,{(FITO/LITO/P)},<Имя блока если не присоед. к цепи>
	UNLINK	UNLINK <Имя цепи>, ...
	ASSIGN	ASSIGN № параметра {_/_+/-/}, Значение,{_ /<Имя функции изен. знач.>}
	INDEX PRIORITY MARK	INDEX № параметра ТА,Значение PRIORITY Приоритетное число,{_/_BUFFER} MARK № параметра ТА

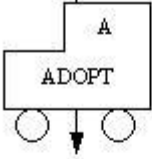
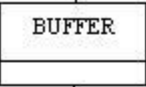
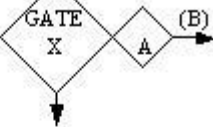
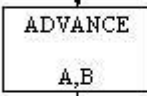
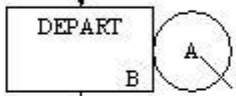
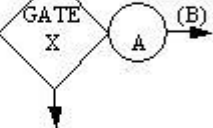
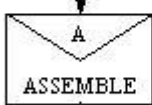
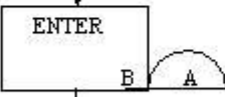
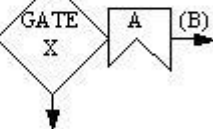
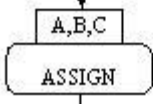
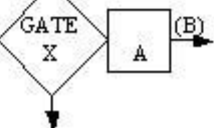

Окончание таблицы 7

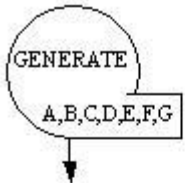
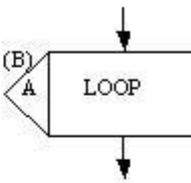
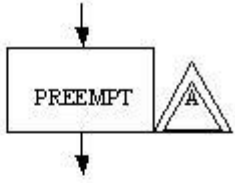
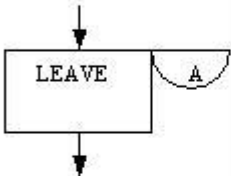
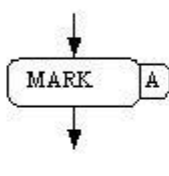
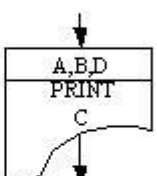
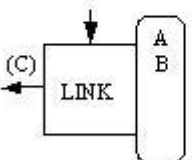
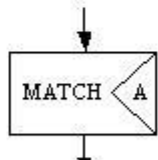
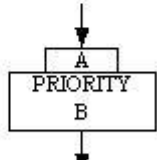
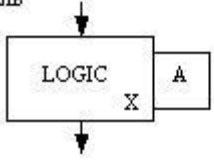
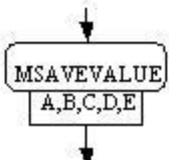
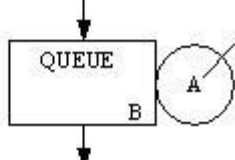
1	2	3
ТА	LINK	LINK <Имя цепи>,{FITO/LITO/PJ},<Имя блока если не присоед. к цепи>
	UNLINK	UNLINK <Имя цепи>, ...
	ASSIGN	ASSIGN № параметра {_/+/-/}, Значение, {_/<Имя функции изен. знач.>}
	INDEX PRIORITY MARK	INDEX № параметра ТА,Значение PRIORITY Приоритетное число,{_/BUFFER} MARK № параметра ТА
Оч.	QUEUE	QUEUE <Имя очереди>,Количество мест
	DEPART	DEPART <Имя очереди>,Количество мест
СВ	INITIAL*	INITIAL <Имя переменной>,Значение
	SAVEVALUE	SAVEVALUE <Имя переменной>{_/+/-/},Значение
	MATRIX*	<Имя матр.> MATRIX_Кол.строк,Кол.столбцов,...,Кол. в 6 измерении
	MSAVE-VALUE	MSAVEVALUE <Имя матрицы>{_/+/-/}, № строки, № столбца, Значение
Таб.	TABLE*	<Имя таблицы> TABLE <Имя переменной>„Левая граница,Ширина интервала,Количество интервалов, {_/Тинт}
	TABULATE	TABULATE <Имя табл.>,{_/Кратность учета}
Пер.	VARIABLE*	<Имя переменной> VARIABLE {Формула/Значение}
	FVARIABLE*	<Имя переменной> FVARIABLE {Формула/Значение}
	BVARIABLE*	<Имя переменной> BVARIABLE Логическое выражение
	EQU*	<Имя переменной> EQU Значение
Фун.	FUNCTION*	<Имя функции> FUNCTION <Имя аргумента>,Количество точек
Прц.	PLUS	PLUS...


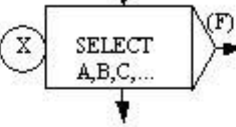
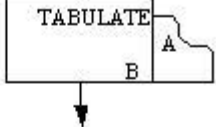
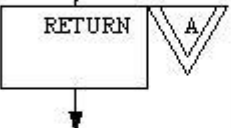
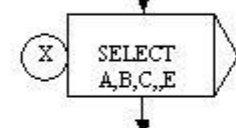

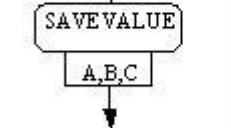

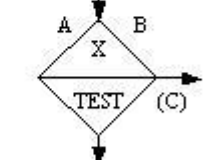
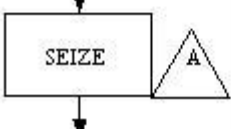
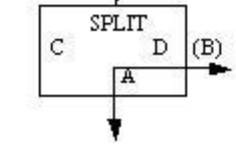
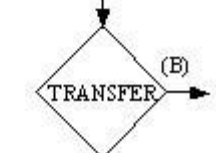
Примечания: Прб. – приборы, Кл. – ключи, Оч. – очереди, СВ – сохраняемые переменные, Таб. – таблицы, Пер. – переменные, Фун. – функции, Прц. – процедуры; * – команды (остальное – блоки); символ ”_” означает отсутствие операнда, а не пробел, т.е. между запятыми пустых позиций нет.

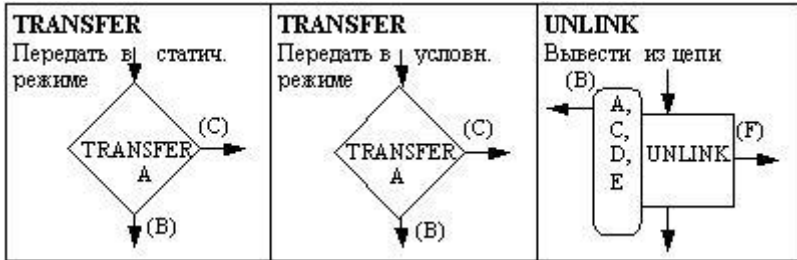
Таблица 8

Условные изображения блоков на блок-схемах GPSS

<p>ADOPT Принять</p> 	<p>BUFFER Возобновить просмотр</p> 	<p>GATE M,NM Впустить</p> 
<p>ADVANCE Задержать</p> 	<p>DEPART Покинуть очередь</p> 	<p>GATE NI,I,NU,U Впустить</p> 
<p>ASSEMBLE Соединить</p> 	<p>ENTER Войти</p> 	<p>GATE SE,SF,SNE,SNF Впустить</p> 
<p>ASSIGN Назначить</p> 	<p>GATE LS,LR Впустить</p> 	<p>GATHER Собрать</p> 

<p>GENERATE Генерировать</p> 	<p>LOOP Организовать цикл</p> 	<p>PREEMPT Захватить</p> 
<p>LEAVE Выйти</p> 	<p>MARK Отметить</p> 	<p>PRINT Напечатать</p> 
<p>LINK Ввести в цепь</p> 	<p>MATCH Синхронизировать</p> 	<p>PRIORITY Назначить приоритет</p> 
<p>LOGIC I,R,S Воздействовать на логический переключатель</p> 	<p>MSAVEVALUE Сохранить значение</p> 	<p>QUEUE Встать в очередь</p> 

<p>RELEASE Освободить</p> 	<p>SELECT U,NU,I,NI, SE,SNE,SF,SNF,LR, LS Выбрать в логи- ческом режиме</p> 	<p>TABULATE Табулировать</p> 
<p>RETURN Вернуть</p> 	<p>SELECT Выбрать в режиме MAX/MIN</p> 	<p>TERMINATE Завершить</p> 
<p>SAVEVALUE Сохранить значение</p> 	<p>SELECT G,GE,E,NE, LE,L Выбрать в режиме отношения</p> 	<p>TEST G,GE,E,NE, LE,L Проверить</p> 
<p>SEIZE Занять</p> 	<p>SPLIT Расщепить</p> 	<p>TRANSFER Передать в безусловном режиме</p> 



4 СЕРВИС СРЕДЫ GPSS

4.1 Отладка и визуализация

Набор текста: Edit / Insert GPSS Blocks...

Управление прогонами:

Первый (образцовый) запуск: Command / Create Simulation.

Перезапуск (без открытия нового журнала): Command / Retranslate.

Временно остановить прогон: Command / Halt.

Продвинуться на шаг (до следующего события): Command / Step1.

Продолжить до выхода очередного ТА: Command / Continue.

Очистка, обнуление и возврат в исходное положение: Command / Clear.

Устранение статистики переходного периода: Command / Reset.

Визуализация:

Увидеть продвижение ТА по блокам:

Window / Simulation Window / Blocs.

Увидеть текущие значения сохраняемых переменных:

Window / Simulation Window / Savevalues.

Увидеть текущие значения выражения (несохраняемой переменной, результата расчета по заданной формуле, СЧА):

Window / Simulation Window / Expression ... ,

Label: <Название>, Expression: {Переменная/формула/СЧА},

<View>, <Ok>.

Увидеть график изменение переменной во времени:

Window / Simulation Window / Plot ...,

Label: <Имя>, Expression: {Переменная/формула/СЧА},

<Plot>.

Time Range (видимый диапазон по оси X –Тмод на графике):

Значение,

Max Value (видимый диапазон по оси Y на графике): Значение,
<Ok>.

Command / Step 1 или Command / Retranslate.

Увидеть гистограмму:

Window / Simulation Window / Table.

Увидеть текущее время, № активного ТА (XN), его местонахождение и следующий по программе блок: в окне "JOURNAL".

Увидеть текущее время моделирования: View / Simulation Clock (появится справа внизу экрана).

4.2 Автоматизация эксперимента

Организация эксперимента и факторный анализ:

Command / Create Simulation,

Edit / Insert Experiment / Screening ...,

Experiment Name: <Имя эксперимента>[SEM],

'Run Procedure' Name: <Имя процедуры>[DoTheRun],

Factors: Name (переменная пользователя):

A: <Имя первой перемен. пользователя >, Value1: Нижн. знач., Value2: Верх.знач.,

B: <Имя второй перемен. пользователя >, Value1: Нижн. знач., Value2: Верх.знач.,

...

F: <Имя шестой перемен. пользователя >, Value1: Нижн. знач., Value2: Верх.знач.,

Fraction: <Full>.

Result Expression (интересующий критерий): {Переменная пользователя/формула/СЧА},

<Insert Experiment>.

Command / Create Simulation / <F11>.

Организация эксперимента и оптимизация:

Command / Create Simulation,

Edit / Insert Experiment / Optimizing...,

Experiment Name: <Имя эксперимента>[RSM],

'Run Procedure' Name: <Имя процедуры>[DoTheRun],

Initial Local Experimental Region (факторное пространство):

Factor Name (переменная пользователя):

A: <Имя первой перем. пользователя >, Value1: Нижн. знач.,

Value2: Верх.знач.,

B: <Имя второй перем. пользователя >, Value1: Нижн. знач., Val-

ue2: Верх.знач.,

...

F: <Имя шестой перем. пользователя >, Value1: Нижн. знач., Val-

ue2: Верх.знач.,

Movement Limits (смещение диапазонов): [0,100],

Redirection Limits (количество опытов в каждой точке плана): [1],

Result Expression (критерий оптимизации):

{Пер.пользователя/формула/СЧА},

<Insert Experiment>,

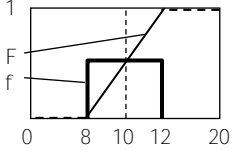
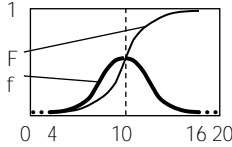
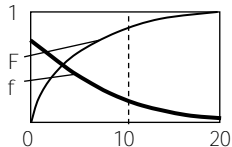
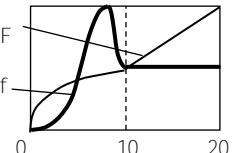
Command / Create Simulation / <F12>.

5 ТИПОВЫЕ КОНСТРУКЦИИ ПРОГРАММ GPSS

Типовые конструкции (фрагменты программ) GPSS приведены в табл. 9.

Таблица 9

Типовые конструкции программ GPSS

A	Ситуация, схема	Текст программы
1	2	3
Генерация ТА	Детерминированная	GENERATE 10 *T = const = 10
	Равномерный закон 	GENERATE 10, 2, 3, 100, 1 *10 – T _{ср} . *2 – T _{размаха} *3 – T _{первой} [T _{ср}] *100 – кол. ТА [∞] *1 – приоритет [0] *T = T _{ср} ± T _{размаха} * (RN) = 10 ± 2
	Нормальный закон 	GENERATE (Normal (2, 10, 2)) *2 – № генератора *10 – T _{ср} *2 – T _{ср.кв.откл.} (σ) *T = T _{ср} ± T _{ср.кв.откл.} * F _{норм} (RN)
	Экспоненциальный закон 	GENERATE (Exponential (2, 0, 10)) *2 – № генератора *0 – T _{min} . *10 – T _{ср} . *T = T _{min} + T _{ср} * F _{эксп} (RN)
	Произвольный закон 	MyZakon FUNCTION RN2, C8 0, 0 / .05, .36 / .1, .52 / .2, .72 / .3, .86 / .4, .88 / .5, 1 / 1, 2 GENERATE 10, FN\$MyZakon *2 – № генератора, *C – вид функции (непрерывная), *8 – количество точек, /X, Y/ – координаты, *10 – T _{ср} *T = T _{ср} * F _{зад} (RN)

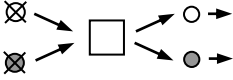

Продолжение таблицы 9

1	2	3
Окончание моделирования	По количеству вышедших ТА (10 шт)	<pre>*-----Вариант 1 (основной) : GENERATE 10,,, SEIZE Prib ADVANCE 12,4 RELEASE Prib TERMINATE 1 START 10 *-----Вариант 2 (без отчета) : GENERATE 10,,,10, SEIZE Prib ADVANCE 12,4 RELEASE Prib TERMINATE 1 START 1000</pre>
	По времени моделирования (100 ед)	<pre>Tmod EQU 100 *-----Вспом. сегмент----- *Единственный фиктивный ТА *с поступлением через Tmod *и мгновенной остановкой GENERATE Tmod TERMINATE 1 START 1 *-----Основной сегмент--- GENERATE 10,2,,, SEIZE Prib ADVANCE 12,4 RELEASE Prib TERMINATE *Оператор TERMINATE без операнда</pre>
	По количеству ТА (50 шт) или по Tмод (100) (что раньше наступит)	<pre>Tmod EQU 100 GENERATE 10,2,,, SEIZE Prib ADVANCE 12,4 RELEASE Prib TEST L AC1,Tmod,Met1 TERMINATE 1 START 50 Met1 REPORT</pre>

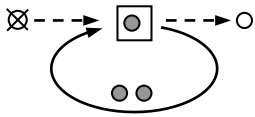
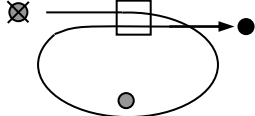

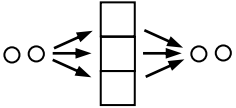
Продолжение таблицы 9

1	2	3
Окончание моделирования	<p>По событию или по их количеству</p>	<pre>Scet EQU 5 Sob EQU 1 GENERATE 10,2,,, SEIZE Prib TEST NE Scet,FC\$Prib,Met2 *Переход в метку 2 после 5 входов в блок ADVANCE 12,4 Met1 RELEASE Prib TEST NE Sob,N\$Met1,Met2 *Переход в метку 2 после входа в блок TERMINATE 1 START 50 Met2 REPORT</pre>
	<p>Партии идентичных ТА</p>	<pre>Part EQU 2 * Партии будут из 3-х ТА (EQU+1) GENERATE 30,,,,,1 *30- периодичность партий SPLIT Part,Met1 Met1 SEIZE Prib</pre>
Источники	<p>Параллельные непересекающиеся</p>	<pre>GENERATE 10,1 SEIZE Prib1 ADVANCE 12,4 RELEASE Prib1 TERMINATE 1 GENERATE 20,2 SEIZE Prib2 ADVANCE 21,6 RELEASE Prib2 TERMINATE 1</pre>
	<p>Пересекающиеся смешивающиеся</p>	<pre>GENERATE 10,1 TRANSFER ,Met1 *Перевод, т.к. в блок GENERATE * ТА никогда не входят GENERATE 20,2 Met1 SEIZE Prib ADVANCE 12,4</pre>


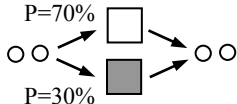
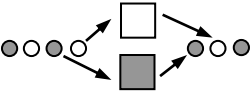
Продолжение таблицы 9

1	2	3
Источники	<p>Пересекающиеся без смешивания (*** – вариант с прерыванием ТА первого потока)</p> 	<pre> *----Вариант 1 (основной) : GENERATE 10,1 SEIZE Prib ADVANCE 10,2 RELEASE Prib TERMINATE 1 GENERATE 20,2 SEIZE Prib *** PREEMPT Prib ADVANCE 20,4 RELEASE Prib *** RETURN Prib TERMINATE 1 START 100 *----Вариант 2 (гибкий) : GENERATE 10,1 ASSIGN 2,Met1 TRANSFER ,Met3 GENERATE 20,2 ASSIGN 2,Met2 TRANSFER ,Met4 Met3 SEIZE Prib ADVANCE 10,2 RELEASE Prib TRANSFER ,P2 Met4 SEIZE Prib ADVANCE 20,4 RELEASE Prib TRANSFER ,P2 Met1 TERMINATE 1 Met2 TERMINATE 1 </pre>
Потоки	<p>Разомкнутый</p> 	<pre> GENERATE 14,2,,,1 SEIZE Prib ADVANCE 12,4 RELEASE Prib TERMINATE 1 </pre>

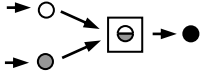
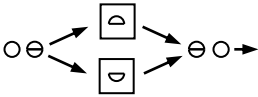
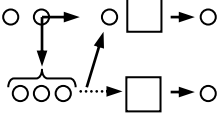
Продолжение таблицы 9

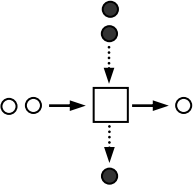
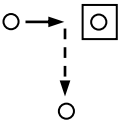
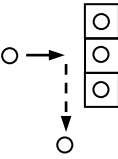
1	2	3
Потоки	<p>Замкнутый</p> 	<p>GENERATE , , , 3, 1 *3 – кол. ТА, циркулирующих внутри Met1 ADVANCE 5, 0 * 5 – Задержка возвращения SEIZE Prib ADVANCE 12, 4 RELEASE Prib SPLIT 1, Met1, 1 TERMINATE 1 *Фактически не вывод ТА, а *регистрация очередного обслуживания</p>
	<p>Разомкнутый, петлевой</p> 	<p>GENERATE 10, 2 ASSIGN 3, 2 *В параметр №3 записать кол. входов = 2 SEIZE Prib Met1 ADVANCE 4, 1 LOOP 3, Met1 RELEASE Prib TERMINATE 1</p>
Маршруты	<p>Последовательный</p> 	<p>GENERATE 10, 1 SEIZE Prib1 ADVANCE 12, 4 RELEASE Prib1 SEIZE Prib2 ADVANCE 8, 2 RELEASE Prib2 TERMINATE 1</p>
	<p>Параллельный, с одинаковыми взаимозаменяемыми приборами</p> 	<p>Grup STORAGE 3 GENERATE 10, 1 ENTER Grup ADVANCE 20, 4 LEAVE Gruppa TERMINATE 1</p>

Продолжение таблицы 9

1	2	3
Маршруты	<p>Параллельный, с предпочтением первому прибору</p> 	<pre> GENERATE 10, 1 TRANSFER BOTH, Met1, Met2 Met1 SEIZE Prib1 ADVANCE 12, 4 RELEASE Prib1 TRANSFER , Met3 Met2 SEIZE Prib2 ADVANCE 8, 2 RELEASE Prib2 Met3 TERMINATE 1 </pre>
	<p>Параллельный, с разной вероятностью</p> 	<pre> GENERATE 10, 1 TRANSFER .3, Met1, Met2 Met1 SEIZE Prib1 ADVANCE 12, 4 RELEASE Prib1 TRANSFER , Met3 Met2 SEIZE Prib2 ADVANCE 8, 2 RELEASE Prib2 Met3 TERMINATE 1 </pre>
	<p>Параллельный, с чередованием приборов (по четным/нечетным ТА)</p> 	<pre> GENERATE 10, 1 LOGIC I Klutc *Инвертирование ключа GATE LS Klutc, Met1 *Проверка: ключ установлен? SEIZE Prib1 ADVANCE 12, 4 RELEASE Prib1 TRANSFER , Met2 Met1 SEIZE Prib2 ADVANCE 8, 2 RELEASE Prib2 Met2 TERMINATE 1 </pre>

Продолжение таблицы 9

1	2	3
Манипуляции с ТА	<p>Сборка ТА из нескольких</p> 	<p>Nak1 STORAGE 1 Nak2 STORAGE 1 GENERATE 18,6 ADOPT 100 ENTER Nak1 TRANSFER ,Met1 GENERATE 18,6 ADOPT 100 ENTER Nak2 Met1 ASSEMBLE 2 SEIZE Prib LEAVE Nak1 LEAVE Nak2 ADVANCE 16,4 RELEASE Prib TERMINATE 1</p>
	<p>Раздельная обработка частей одного ТА</p> 	<p>GENERATE 18,6 SPLIT 1, Met1 SEIZE Prib1 ADVANCE 16,4 RELEASE Prib1 TRANSFER ,Met2 Met1 SEIZE Prib2 ADVANCE 5,1 RELEASE Prib2 Met2 ASSEMBLE 2 TERMINATE 1</p>
	<p>Размножение ТА (***) – вариант для маршрута по пунктирной стрелке)</p> 	<p>GENERATE 20 SPLIT 3, *** SPLIT 3, Met1 SEIZE Prib1 ADVANCE 1 RELEASE Prib1 Met2 TERMINATE 1 START 100 Met1 SEIZE Prib2 ADVANCE 2 RELEASE Prib2 TRANSFER ,Met2</p>

1	2	3
<p style="writing-mode: vertical-rl; transform: rotate(180deg);">Нештатные ситуации</p>	<p>Поломки и ремонты</p> 	<pre>*----Сегмент1 (основной) - GENERATE 10,1 SEIZE Prib ADVANCE 10,2 RELEASE Prib TERMINATE 1 START 100 *----Сегмент2 (отказы) -- GENERATE 100,2 * PREEMPT Prib ADVANCE 20,4 * RETURN Prib TERMINATE</pre>
	<p>Отказы в обслуживании из-за занятости</p> 	<pre>*----Сегмент для занятости GENERATE 10,4 GATE NU Prib,Fin SEIZE Prib ADVANCE 10,4 RELEASE Prib Fin TERMINATE 1</pre>
	<p>Отказы в обслуживании из-за переполнения</p> 	<pre>Nak STORAGE 2 GENERATE 10,4 GATE SNF Nak,Fin ENTER Nak,1 SEIZE Prib LEAVE Nak,1 ADVANCE 10,4 RELEASE Prib Fin TERMINATE 1</pre>

Примечания: А – аспект; * – комментарии к предыдущей строке; [...] – значения по умолчанию.

СОДЕРЖАНИЕ

Введение.....	3
1. ОСНОВНЫЕ ПОЛОЖЕНИЯ GPSS.....	3
1.1 Назначение языка	3
1.2 Пример модели	4
1.3 Объекты языка	5
1.4 Формализация объекта моделирования.....	8
1.5 Принцип построения программы	9
1.6 Принцип имитации	10
1.7 Цепи событий	13
1.8 Атрибуты объектов	14
1.9 Ансамбли транзактов	15
2 СИНТАКСИС ЯЗЫКА GPSS	16
2.1 Алфавит	16
2.2 Числа	16
2.3 Формат строки программы	16
2.4 Операторы	17
2.5 Стандартные числовые атрибуты (СЧА).....	19
2.6 Методы адресации	21
3 БЛОКИ И КОМАНДЫ GPSS	22
4 СЕРВИС СРЕДЫ GPSS	31
4.1 Отладка и визуализация	31
4.2 Автоматизация эксперимента	32
5 ТИПОВЫЕ КОНСТРУКЦИИ ПРОГРАММ GPSS	33

Учебное издание

НОВИЧИХИНА Елена Романовна
НОВИЧИХИН Роман Васильевич

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ
В СРЕДЕ GPSS

Методическое пособие
по дисциплине
«Моделирование систем»
для студентов специальности
1-53 01 02 «Автоматизированные системы обработки
информации»

Редактор _____
Компьютерная верстка _____

Подписано в печать __.__.2010
Формат 60x84 1/16. Бумага офсетная.
Отпечатано на ризографе. Гарнитура Таймс.
Усл. печ. л. ____. Уч.-изд. л. ____. Тираж 100. Заказ ____

Издатель и полиграфическое исполнение:
Белорусский национальный технический университет.
ЛИ № 02330/0131627 от 01.04.2004.
220013, Минск, проспект Независимости, 65.