

Язык GPSS.

Изменение маршрутов транзактов.

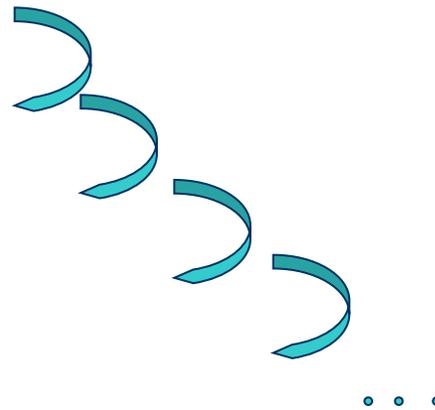
Вычислительные объекты языка .

Лектор:
доцент каф. АОИ
Салмина Нина
Юрьевна

Обычное путешествие транзактов по модели

Транзакты переходят последовательно от блока к блоку:

GENERATE
QUEUE
SEIZE
DEPART
ADVANCE
RELEASE
TERMINATE



Безусловные переходы

TRANSFER A,B,C,D

A – режим перехода;

- 1) A – пусто: *режим безусловного перехода*
B – имя блока, куда направляется транзакт.

Пример:

```
generate 5,2  
...  
transfer ,prod  
...  
prod terminate
```



Статистические переходы

TRANSFER A,B,C,D

2) A – real на интервале $(0,1)$: **режим статистического перехода**

Внимание! 0 и 1 не использовать!

A – вероятность

B, C – имена блоков

транзакт с $P(A)$ – переход по метке C

с $P(1-A)$ – переход по метке B ,
если B пусто, то переход
к следующему блоку

Пример статистического перехода

Задача:

В магазине работает два отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

40% покупателей направляется в 1-й отдел

60% покупателей – во 2-й отдел

Модель:

```
generate (exponential(1,0,5))
transfer .6,otd1,otd2
otd1 queue och1
seize prod1
...
terminate
otd2 queue och2
seize prod2
...
terminate
```

Пример статистического перехода

Задача:

В магазине работает два отдела.

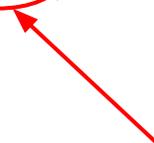
Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

40% покупателей направляется в 1-й отдел

60% покупателей – во 2-й отдел

Модель:

```
generate (exponential(1,0,5))
transfer .6,otd1,otd2
otd1 queue och1
      seize prod1
...
terminate
otd2 queue och2
      seize prod2
...
terminate
```



Можно опустить

Пример статистического перехода (три разветвления)

Задача:

В магазине работает три отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

40% покупателей направляется в 1-й отдел

30% покупателей – во 2-й отдел

30% покупателей – в 3-й отдел

Модель:

```
generate (exponential(1,0,5))  
transfer .4,,otd1
```

???

Пример статистического перехода (три разветвления)

Задача:

В магазине работает три отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

40% покупателей направляется в 1-й отдел

30% покупателей – во 2-й отдел

30% покупателей – в 3-й отдел

Модель:

```
generate (exponential(1,0,5))
transfer .4,,otd1
transfer .5,,otd2
otd3 queue och3
...
terminate
otd1 queue och1
...
terminate
otd2 queue och2
...
terminate
```

Пример статистического перехода (большое количество разветвлений)

Задача:

В магазине работает четыре отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

21% покупателей направляется в 1-й отдел

17% покупателей – во 2-й отдел

34% покупателей – в 3-й отдел

28% покупателей – в 4-й отдел

Модель:

```
generate (exponential(1,0,5))  
transfer .21,,otd1
```

↓ проходит 79%

```
transfer
```

```
transfer
```

```
otd4 queue och4
```

```
...
```

```
otd1 queue och1
```

```
...
```

Пример статистического перехода (большое количество разветвлений)

Задача:

В магазине работает четыре отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

21% покупателей направляется в 1-й отдел

17% покупателей – во 2-й отдел

34% покупателей – в 3-й отдел

28% покупателей – в 4-й отдел

Модель:

```
generate (exponential(1,0,5))  
transfer .21,,otd1
```

↓ проходит 79%

```
transfer .2152,,otd2 ;17/79
```

↓ проходит 62%

```
transfer .5484,,otd2 ;34/62
```

↓ проходит 28%

```
otd4 queue och4
```

...

```
otd1 queue och1
```

...

Пример статистического перехода (использование функции)

Задача:

В магазине работает четыре отдела.

Покупатели приходят в среднем каждые 5 минут (экспоненциальный закон распределения)

21% покупателей направляется в 1-й отдел

17% покупателей – во 2-й отдел

34% покупателей – в 3-й отдел

28% покупателей – в 4-й отдел

Модель:

Perehod function rn1,D4

.21,otd1/.38,otd2/.72,otd3/1,otd4

generate (exponential(1,0,5))

transfer ,fn\$Perehod

otd1 queue och1

...

otd2 queue och2

...

otd3 queue och3

...

otd4 queue och4

...

Проверка состояния оборудования

GATE O A,B

O – мнемокод состояния оборудования

A – имя оборудования

B – имя блока

Оборудование:	устройство	память
Мнемокод	NU – свободно	SE – пуста
состояния:	U – занято	SNE – не пуста
	SF – полна	
	SNF – не полна	

Проверка состояния оборудования (продолжение)

GATE O A,B

Режимы работы:

1) В – пусто: *режим задержки*

если оборудование **A** находится в состоянии **O** – переход к следующему блоку

если не находится – задержка до тех пор, пока оборудование не окажется в состоянии **O**

2) В – имя блока: *режим условного перехода*

если оборудование **A** находится в состоянии **O** – переход к следующему блоку

если не находится – переход по метке **B**

Истина

Ложь



Пример 1 (постановка задачи)

В цеху по изготовлению деталей работают мастер и ученик.
Поступают заготовки деталей 2-х типов.

1-й тип. Время поступления 10 ± 3 мин

если мастер свободен – ему, иначе – ученику.

Мастер тратит на изготовление 9 ± 2 мин

Ученик тратит на работу 15 ± 5 мин.

После ученика мастер проверяет деталь в течение 1 ± 0.5 мин.

В результате: 85% - норма,

10% - доделка (мастером) в течение 3 ± 2 мин,

5% - брак.

2-й тип. Время поступления 30 ± 10 мин

выполняет **только** мастер в течение 20 ± 3 мин

Промоделировать в течение 8 часов.

Пример 1 (модель)

```
rab_uch function m1,d3  
.05,brak/.15,dod/1,norm
```

```
generate 30,10  
seize mast  
advance 20,3  
release mast  
terminate
```

Обработка
мастером деталей
2-го типа

Пример 1 (модель)

```
rab_uch function rn1,d3  
.05,brak/.15,dod/1,norm
```

```
generate 30,10  
seize mast  
advance 20,3  
release mast  
terminate
```

Обработка
мастером деталей
2-го типа

```
generate 10,3  
gate nu mast,k_uch  
seize mast  
advance 9,2  
release mast  
terminate
```

Обработка
мастером
деталей 1-го
типа

Пример 1 (модель)

```
rab_uch function rn1,d3  
.05,brak/.15,dod/1,norm
```

```
generate 30,10  
seize mast  
advance 20,3  
release mast  
terminate
```

Обработка
мастером деталей
2-го типа

```
generate 10,3  
gate nu mast,k_uch  
seize mast  
advance 9,2  
release mast  
terminate
```

Обработка
мастером
деталей 1-го
типа

```
k_uch queue uch  
seize uch  
depart uch  
advance 15,5  
release uch  
priority 1  
seize mast  
advance 1,0.5  
transfer ,fn$rab_uch
```

Работа
ученика

```
norm release mast  
terminate  
brak release mast  
terminate  
dod advance 3,2  
release mast  
terminate
```

Результат
работы
ученика

Пример 1 (модель)

```
rab_uch function rn1,d3  
.05,brak/.15,dod/1,norm
```

```
generate 30,10  
seize mast  
advance 20,3  
release mast  
terminate
```

Обработка
мастером деталей
2-го типа

```
generate 10,3  
gate nu mast,k_uch  
seize mast  
advance 9,2  
release mast  
terminate
```

Обработка
мастером
деталей 1-го
типа

```
generate 480  
terminate 1
```

Таймер

```
k_uch queue uch  
seize uch  
depart uch  
advance 15,5  
release uch  
priority 1  
seize mast  
advance 1,0.5  
transfer ,fn$rab_uch
```

Работа
ученика

```
norm release mast  
terminate  
brak release mast  
terminate  
dod advance 3,2  
release mast  
terminate
```

Результат
работы
ученика

Пример 1 (отчет)

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY	
	1	GENERATE	14	0	0		
	2	SEIZE	14	0	0		
	3	ADVANCE	14	1	0		← Мастер обрабатывает деталь 2-го типа
	4	RELEASE	13	0	0		
	5	TERMINATE	13	0	0		
	6	GENERATE	48	0	0		
	7	GATE	48	0	0		
	8	SEIZE	10	0	0		
	9	ADVANCE	10	0	0		
	10	RELEASE	10	0	0		
	11	TERMINATE		10	0	0	7 деталей в очереди к ученику
K_UCH	12	QUEUE		38	7	0	← 0
	13	SEIZE	31	0	0		
	14	DEPART	31	0	0		
	15	ADVANCE	31	1	0		← Ученик обрабатывает деталь
	16	RELEASE	30	0	0		
	17	PRIORITY	30	1	0		← Деталь 1-го типа в очереди к мастеру на проверку
	18	SEIZE	29	0	0		
	19	ADVANCE	29	0	0		
	20	TRANSFER		29	0	0	
NORM	21	RELEASE		25	0	0	
	22	TERMINATE		25	0	0	
BRAK	23	RELEASE		0	0	0	
	24	TERMINATE		0	0	0	
DOD	25	ADVANCE		4	0	0	
	26	RELEASE		4	0	0	
	27	TERMINATE		4	0	0	
	28	GENERATE		1	0	0	
	29	TERMINATE		1	0	0	

Пример 1 (отчет продолжение)

FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
MAST	53	0.888	8.047	1	58	0	0	0	1
UCH	31	0.915	14.166	1	53	0	0	0	7

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
UCH	7	7	38	1	3.763	47.529	48.814	0

Стандартные числовые атрибуты (СЧА)

Любой объект языка имеет свои СЧА

Обращение к СЧА:

<СЧА><номер объекта>

<СЧА>\$<имя объекта>

Примеры:

RN1, RN45 – значение датчиков случайных чисел с равномерным распределением

FN\$Perehod – значение функции с именем Perehod

СЧА блоков, устройств, памятей

Блоки

СЧА	Значение
N\$<имя бл>	Число Тр, вошедших в блок
W\$<имя бл>	Число Тр, находящихся в блоке

Устройство

СЧА	Значение
FC\$<имя У>	Число Тр, вошедших в Устр
FT\$<имя У>	Среднее время занятия одним Тр устройства

Память

СЧА	Значение
S\$<имяП>	текущее содержимое П
R\$<имяП >	Свободный объем П
SA\$<имяП >	Среднее число занятых каналов
SC\$<имяП >	Число Тр, вошедших в П
SM\$<имяП >	Макс.кол-во занятых ед. П
ST\$<имяП >	Среднее время нахождения одного Тр в П

СЧА статистических очередей

Память

СЧА	Значение
Q\$<имяП>	Текущая длина Оч
QA\$<имяП >	Средняя длина очереди
QC\$<имяП >	Число Тр, вошедших в Оч
QM\$<имяП >	Макс.длина Оч
QZ\$<имяП>	Количество нулевых входов
QT\$<имяП >	Среднее время нахождения одного Тр в Оч
QX\$<имяП>	Среднее время нахождения Тр в Оч без нулевых вх

Примеры использования СЧА:

LEAVE PAM,\$\$PAM – освободить память
полностью

SEIZE FN\$USTR – занять устройство,
имя которого
определено
функцией USTR

Блок условного перехода

TEST O A,B,C

O – мнемоника проверяемого отношения

A,B – левое и правое значения условия, заданные в виде СЧА

C – имя блока

Мнемоники отношений:

L – “<”

E – “=”

G – “>”

LE – “≤”

NE – “≠”

GE – “≥”

Блок условного перехода

TEST O A,B,C

Режимы работы:

1) **C** – пусто: *режим задержки*

если **A** находится с **B** в заданном отношении **O** – переход к следующему блоку (**A O B**)

если не находится – задержка до тех пор, условие **O** не выполнится

2) **C** – имя блока: *режим условного перехода*

если **A** находится с **B** в заданном отношении **O** – переход к следующему блоку

если не находится – переход по метке **C**

Истина

Ложь



Переменные

<имя> VARIABLE A

A – арифметическое выражение (из любых СЧА)

«+», «-», «/»,

«#» – умножение, «\» – целое от деления

СЧА: **V\$<имя переменной>**

Примеры:

Общее время простоя в очереди OCH

PROSTOI VARIABLE QC\$OCH#QTOCH

Содержание каналов STR за 10 часов (стоимость 1 кан/час=245)

SUM VARIABLE (S\$STR+R\$STR)#245#10

Ячейки

SAVEVALUE A[±],B

A – имя/номер ячейки

B – присваиваемое значение (константа, СЧА)

A,B – ячейке с именем A присвоить значение B

A+,B – в ячейку с именем A добавить значение B

A-,B – из ячейки с именем A вычесть значение B

(начальное значение любой ячейки равно 0)

СЧА: **X\$<имя ячейки>** или **X<№ячейки>**

Примеры:

В ячейку с номером 10 записать значение переменной PROSTOI

SAVEVALUE 10,V\$PROSTOI

Добавить в ячейку SUMMA 10 единиц

SAVEVALUE SUMMA+,10

Пример 2 (постановка задачи)

В магазине работает два отдела:

В 1-м отделе работает 2 продавца, время обслуживания – 7 минут;
если очередь к продавцам скапливается больше 2-х человек, то очередной покупатель уходит.

Во 2-м отделе работает 1 продавец, время обсл. – 3 мин, очередь не ограничена.

Время прихода покупателей равно в среднем 2 мин, причем примерно 70% покупателей направляется в 1-й отдел, а 30% - во второй.

Стоимость покупки (равномерный закон распределения):

1-й отдел: от 200 до 3000 рублей

2-й отдел: от 100 до 1000 рублей

Все временные хар-ки подчиняются экспоненциальному закону распределения.

Общее время моделирования – 8 часов работы магазина * 30 дней

Определить: прибыль магазина за месяц.

Пример 2 (модель)

;прибыль от покупки

Prib1 function rn1,c2

0,200/1,3000

Prib2 function rn1,c2

0,100/1,1000

;общая прибыль магазина

Osum variable x\$sum1+x\$sum2

Prod1 storage 2

Пример 2 (модель)

;прибыль от покупки

Prib1 function rn1,c2

0,200/1,3000

Prib2 function rn1,c2

0,100/1,1000

;общая прибыль магазина

Osum variable x\$sum1+x\$sum2

Prod1 storage 2

;приход покупателей

GENERATE (exponential(1,0,2))

;распределение по отделам

Transfer .3,otd1,otd2

Пример 2 (модель)

;прибыль от покупки

Prib1 function rn1,c2

0,200/1,3000

Prib2 function rn1,c2

0,100/1,1000

;общая прибыль магазина

Osum variable x\$sum1+x\$sum2

Prod1 storage 2

;приход покупателей

GENERATE (exponential(1,0,2))

;распределение по отделам

Transfer .3,otd1,otd2

;работа 1-го отдела

Otd1 TEST LE Q\$Och1,2,Uhod

QUEUE Och1

ENTER Prod1

DEPART Och1

ADVANCE (exponential(1,0,7))

LEAVE Prod1

SAVEVALUE sum1+,FN\$Prib1

Uhod TERMINATE

;работа 2-го отдела

Otd2 QUEUE Och2

SEIZE Prod2

DEPART Och2

ADVANCE (exponential(1,0,3))

RELEASE Prod2

SAVEVALUE sum2+,FN\$Prib2

TERMINATE

Пример 2 (модель)

;прибыль от покупки

Prib1 function rn1,c2

0,200/1,3000

Prib2 function rn1,c2

0,100/1,1000

;общая прибыль магазина

Osum variable x\$sum1+x\$sum2

Prod1 storage 2

;приход покупателей

GENERATE (exponential(1,0,2))

;распределение по отделам

Transfer .3,otd1,otd2

;таймер

GENERATE (8#60#30)

SAVEVALUE itog,V\$Osum

TERMINATE 1

START 1

;работа 1-го отдела

Otd1 TEST LE Q\$Och1,2,Uhod

QUEUE Och1

ENTER Prod1

DEPART Och1

ADVANCE (exponential(1,0,7))

LEAVE Prod1

SAVEVALUE sum1+,FN\$Prib1

Uhod TERMINATE

;работа 2-го отдела

Otd2 QUEUE Och2

SEIZE Prod2

DEPART Och2

ADVANCE (exponential(1,0,3))

RELEASE Prod2

SAVEVALUE sum2+,FN\$Prib2

TERMINATE

Пример 2 (модель)

;прибыль от покупки

Prib1 function rn1,c2

0,200/1,3000

Prib2 function rn1,c2

0,100/1,1000

;общая прибыль магазина

Osum variable x\$sum1+x\$sum2

Prod1 storage 2

;приход покупателей

GENERATE (exponential(1,0,2))

;распределение по отделам

Transfer .3,otd1,otd2

;таймер

GENERATE (8#60#30)

SAVEVALUE itog,V\$Osum

TERMINATE 1

START 1

;работа 1-го отдела

Otd1 TEST LE Q\$Och1,2,Uhod

QUEUE Och1

ENTER Prod1

DEPART Och1

ADVANCE (exponential(1,0,7))

LEAVE Prod1

SAVEVALUE sum1+,FN\$Prib1

Uhod TERMINATE

;работа 2-го отдела

Otd2 QUEUE Och2

SEIZE Prod2

DEPART Och2

ADVANCE (exponential(1,0,3))

RELEASE Prod2

SAVEVALUE sum2+,FN\$Prib2

TERMINATE

Пример 2 (результаты моделирования)

SAVEVALUE	RETRY	VALUE
SUM1	0	5827987.155
SUM2	0	1165347.805
ITOG	0	6993334.960