

Тема 13.

Збережувані величини і матриці. Змінні та вирази.

Зміна маршрутів транзактів

13.1. Збережувані величини і матриці

Транзакти не можуть безпосередньо посилатися один на одного. Їхнє спілкування реалізується через збережувані величини. Збережувані величини можуть бути скалярними і матричними.

Блоки, які змінюють значення параметрів транзактів

Кожен транзакт може мати будь-яку кількість параметрів з довільною інтерпретацією суті кожного з них. У момент генерування транзакта всі його параметри (які використовують в моделі) мають нульові значення. Для присвоювання параметрам початкових значень або зміни цих значень призначений блок ASSIGN, який має такий формат:

ASSIGN A,B,C

Операнд A задає номер параметра, якому присвоюється значення. Він може бути іменем, додатним цілим числом, виразом у дужках, СЧА, СЧА*параметром. Операнд A може супроводжуватися знаком "+" або "-" (якщо треба збільшити або зменшити значення параметра транзакта, вказаного в операнді A).

Операнд B визначає значення, яке потрібно додати, відняти або яким треба замінити значення в параметрі, заданому операндом A. Якщо такого параметра немає, то він автоматично створюється зі значенням, рівним 0. Операнд B може бути таким самим, як і операнд A, крім того, числом і рядком.

Операнд C (необов'язковий) задає номер або ім'я функції. У випадку використання операнда C значення операнда B множиться на значення функції. Отриманий результат множення стає значенням, яке змінює значення параметра, заданого операндом A. Оскільки операнд C визначає номер функції або її ім'я, то не потрібно використовувати СЧА FN перед ним. Функція в цьому випадку є *функцією-модифікатором*. Крім того, операнд C може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Приклади запису блока ASSIGN:

ASSIGN 5,23.2

ASSIGN 3+,Q\$Cherga

ASSIGN 4-,33.5,7
ASSIGN Imen,"Klon"
ASSIGN Chas-, (X5+Q5),Nord

У першому прикладі параметру номер 5 присвоюється число 23,2. У другому прикладі до значення параметра 3 додається значення поточної довжини черги з ім'ям Cherga. У третьому прикладі від значення параметра номер 4 віднімається результат множення числа 33,5 на обчислене значення функції номер 7. У четвертому прикладі параметру з іменем Imen присвоюється рядок Klon. У п'ятому прикладі обчислюються вираз у дужках і функція з іменем Nord, і результат їхнього добутку віднімається від значення параметра з іменем Chas.

У GPSS World значення параметрів транзактів можна також змінювати за допомогою блока PLUS. Блок PLUS обчислює вираз і записує його в параметр. Блок PLUS має такий формат:

PLUS A,B

Операнди:

A – вираз, який може бути ім'ям, числом, рядком, виразом у дужках, СЧА або СЧА*параметром;

B — номер параметра транзакта, в якому зберігається результат. Може бути ім'ям, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Наприклад:

PLUS (P\$Result+1.25#Q4),Res

Коли транзакт потрапляє у блок PLUS обчислюється вираз у дужках, записаний у полі операнда A. Потім отриманий результат присвоюється параметру з іменем Res. Якщо такого параметра немає, то він автоматично створюється.

Змінити значення параметра транзакта можна також за допомогою блока INDEX. Формат блоку:

INDEX A,B

Операнди:

A – задає номер параметра транзакта, який може бути ім'ям, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Операнд B — числове значення, яке додається до значення параметра, вказаного операндом A. Може бути іменем, числом, виразом у дужках, СЧА або СЧА*параметром.

Отриманий результат заноситься в перший параметр активного транзакта.

Наприклад:

INDEX Dod,(Normal(3,Met,Ser)+X\$Ost)

Після потрапляння транзакта у блок INDEX обчислюється вираз у дужках, записаний у полі операнда B, і результат додається до значення параметра з іменем Dod. Отриманий результат присвоюється параметру номер 1 транзакта. Якщо цього параметра не існує, то він автоматично створюється. Якщо немає параметра, зазначеного у полі операнда A, то відбувається зупинка внаслідок помилки "Звертання до неіснуючого параметра".

Блок MARK використовують для занесення до активного транзакта або його заданого параметра значення абсолютного модельного часу. Цей блок має такий формат:

MARK A

Операнд A — номер параметра транзакта, в який автоматично записується значення абсолютного модельного часу в момент потрапляння цього транзакта у блок MARK. Може бути ім'ям, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

MARK

MARK Vhid

У першому прикладі операнд A не використовується. За замовчуванням транзакту, що увійшов в цей блок, встановлюється поточний час замість часу надходження транзакта до моделі. Цей поточний час використовують для визначення СЧА M1. СЧА M1 обчислюється як різниця значення абсолютного модельного часу (СЧА AC1) і часу, присвоєного транзакту, який дорівнює або часу надходження транзакта до моделі, або часу останнього проходження транзактом блока MARK без операнда A.

У другому прикладі операнд A використовується. Тому значення абсолютного модельного часу заноситься в параметр з іменем Vhid транзакта, який увійшов до цього блока MARK. Якщо цього параметра немає, то він автоматично створюється.

Блок MARK можна використовувати, якщо необхідно визначити час проходження транзакта через певну ділянку моделі. Для цього використовують СЧА MP. Наприклад, якщо на вході ділянки моделі розташувати блок

MARK Marker

то на виході цієї ділянки СЧА MP\$Marker буде містити різницю між поточним модельним часом і часом, занесеним у параметр Marker блоком MARK.

Зауважимо, що у блоках ASSIGN, PLUS, INDEX і MARK під час запису в поля операндів номера або ім'я параметра транзакта СЧА P не

використовується.

У параметр транзакта можна записати результат вибору об'єктів за умовою або кількістю об'єктів, що відповідають заданій умові. Для цього використовують блоки SELECT і COUNT.

SELECT X A,B,C,D,E,F

Блок вибирає об'єкт за умовою й поміщає його номер у параметр активного транзакта. Операнди:

X – логічний оператор або показчик відношення;

A – операнд для запису номера обраного елемента; B – нижня межа діапазону пошуку;

C – верхня межа діапазону пошуку;

D – опорне значення (база порівняння) з операндом E;

E – ім'я класу СЧА, обов'язковий тільки для умовної форми;

F – номер блока призначення, якщо не відібраний жоден об'єкт.

Приклади:

```
SELECT LE 3,1,10,.70,FR
```

```
SELECT min Minicherga,1,7, ,Q
```

```
SELECT FV TTT,1,5,,,Mit1
```

У першому прикладі вибирається з 10 пристроїв той, у якого коефіцієнт використання менше або дорівнює .70, номер пристрою (тобто результат вибору) записується в 3-й параметр активного транзакта. У другому прикладі вибирається найкоротша черга з семи існуючих і записується її номер у параметр транзакта з ім'ям Minicherga. У третьому прикладі вибирається доступний пристрій з 5-ти можливих і записується його номер у параметр TTT активного транзакта. Якщо доступних пристроїв не виявиться, транзакт направляється до блока з міткою Mit1.

COUNT X A,B,C,D,E

Блок підраховує число об'єктів, що відповідають заданій умові, і записує результат у параметр активного транзакта. Операнди:

X – логічний оператор або показчик відношення; A

– операнд для прийняття результату;

B – нижня межа діапазону об'єктів, що перевіряються; C – верхня межа діапазону;

D – опорне значення для операнда E, обов'язкове тільки при наявності умови;

E – специфікатор класу об'єктів (для умовного режиму перевірки).

Приклади:

COUNT LE 3,1,10,.70,FR

COUNT E Minicherga,1,7,12,Q

COUNT FV TTT,1,5

У першому прикладі в 3-й параметр активного транзакта записується кількість пристроїв з 10 можливих, у яких коефіцієнт використання менше або дорівнює .70. У другому прикладі в параметр з ім'ям Minicherga записується кількість черг (з 7-ми можливих), які містять 12 транзактів. У третьому прикладі підраховується кількість доступних пристроїв з 5-ти можливих і записується результат у параметр TTT активного транзакта.

Комірки зберігання

У GPSS використовують комірки пам'яті, початкові значення яких можуть бути задані перед моделюванням і до яких можна звернутися з будь-якого місця моделі під час моделювання. Ці комірки називають **комірками зберігання**. Значення комірок зберігання змінюються лише за прямою вказівкою користувача. Перед початком моделювання встановлюють значення комірок зберігання, які дорівнюють нулю. За бажанням користувача окремим коміркам зберігання можуть бути присвоєні ненульові початкові значення за допомогою команди INITIAL. Команду INITIAL застосовують за відсутності активного транзакта або ж тоді, коли не можна використати значення параметрів транзактів. Вона має такий формат:

INITIAL A,B

Операнд A під час ініціалізації комірки зберігання може бути X додатне_ціле_число, X\$<ім'я>. Тобто в команді можна відразу зазначати або ім'я або номер збережуваної величини.

Операнд B — початкове значення, яке привласнюється збережуваній величині, або UNSPECIFIED (не визначене). Може бути іменем, числом, рядком або UNSPECIFIED. Якщо операнд B не використовують, то значення комірки дорівнює 1.

Наприклад:

INITIAL X7,123.45

INITIAL X\$Res,Q\$Cherga

INITIAL X\$Komirka,V\$Myzmin

INITIAL X\$Chas,"Rezult"

INITIAL X4

У першому прикладі в комірку номер 7 заноситься число 123.45, у другому — в комірку з іменем Res записується поточна довжина черги з іменем Cherga, у третьому – в комірку з іменем Komirka – значення змінної користувача Myzmin, у четвертому — в комірку з іменем Chas заноситься рядкова константа Rezult, а в п'ятому прикладі операнд В не використовується, тому комірці зберігання номер 4 присвоюється значення 1.

У процесі моделювання значення комірки зберігання змінюється з входженням транзакта у блок SAVEVALUE (зберегти величину). Формат запису блоку SAVEVALUE:

SAVEVALUE A,B

Операнд А задає номер або ім'я збережуваної величини. Він може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Операнд А може супроводжуватися знаком "+" або "-" (якщо треба збільшити або зменшити значення збережуваної величини, вказаної в операнді А).

Операнд В визначає значення, яке потрібно додати, відняти або яким треба замінити значення збережуваної величини, заданої операндом А. Якщо такої збережуваної величини немає, то вона автоматично створюється зі значенням, рівним нулю. Операнд В може бути таким самим, як і операнд А, крім того, числом і рядком.

Приклади:

SAVEVALUE	6+,X3
SAVEVALUE	Dva-,V\$Imc
SAVEVALUE	Tret,-18
SAVEVALUE	4, (3.7#X\$Nom+Q6)
SAVEVALUE	15,"Rezult number..."
SAVEVALUE	P9,V\$Zmin

Значення операндів А змінюються лише із входженням транзактів до цих блоків. У першому прикладі значення комірки зберігання номер 6 з входженням транзакта у блок SAVEVALUE збільшується на значення комірки зберігання з номером 3. У другому прикладі значення комірки зберігання з іменем Dva зменшується на обчислене значення арифметичної змінної Imc. У третьому прикладі значення комірки зберігання з іменем Tret заміщується числом –18. У четвертому прикладі обчислюється вираз у дужках і присвоюється комірці номер 4. У п'ятому вміст комірки номер 15 заміщується рядком. Якщо комірки з таким іменем не

існує, то вона створюється. У шостому прикладі обчислюється значення змінної користувача *Zmin*, отриманий результат призначається комірці зберігання, номер якої записаний у параметрі номер 9 активного транзакта (одночасно старе значення комірки зберігання знищується).

Комірки зберігання мають єдиний СЧА з назвою *X*, значенням якого є поточне значення відповідної комірки зберігання. СЧА *X* в операнді *A* блока *SAVEVALUE*, як видно з прикладів, не використовується. Під час запису інших операндів посилання на СЧА *X* обов'язкове. Посилання на цей СЧА застосовують також у виразах, функціях, в команді *INITIAL*.

Матриці комірок зберігання

Поняття комірки зберігання в GPSS застосовують і до матриць. Ці об'єкти дають змогу впорядкувати значення, які потрібно зберегти у вигляді матриць. Матрицю необхідно спочатку описати. Команда опису матриці має такий формат:

<ім'я матриці> MATRIX A,B,C,D,E,F,G

Мітка <ім'я матриці> визначає ім'я матриці і повинне бути іменем (не може бути числом). Операнд *A* не використовують, його залишено для сумісності зі старішими версіями GPSS. Операнди *B*, *C*, *D*, *E*, *F*, *G* можуть бути лише цілими додатними числами.

Операнд *B* задає кількість рядків матриці — максимальну кількість елементів у першому вимірі, операнд *C* — кількість стовпців матриці — максимальну кількість елементів у другому вимірі. Операнди *D*, *E*, *F*, *G* задають максимальну кількість елементів у третьому, четвертому, п'ятому і шостому вимірах відповідно. Наприклад, команда

Matr MATRIX ,5,7

визначає матрицю з іменем *Matr*, яка містить 5 рядків і 7 стовпців.

Матриці мають єдиний СЧА з назвою *MX*, за допомогою якого мож-а звертатися до будь-якого елемента матриці. Значеннями рядків і стовпців можуть бути імена, цілі числа і параметри транзактів. Приклади:

MX1(2,5)

MX\$Tab(P3,P\$Stovp)

MX\$Tab(V\$Ser,P4)

У першому прикладі визначається елемент матриці номер 1, який перебуває на перетині другого рядка і п'ятого стовпця. У другому прикладі для визначення елемента матриці з іменем *Tab* використовуються значення параметра номер 3 і параметра з іменем *Stovp* транзакта, а у третьому — ім'я змінної користувача з іменем *Ser* і параметр номер 4 транзакта.

Зауважимо, що у першому прикладі матриця має не ім'я, а номер, хоч у команді визначення матриці, як було зазначено вище, на місці мітки <ім'я матриці> можна використовувати лише ім'я. Для присвоєння матриці номера замість імені потрібно для запису команди визначення матриці використати команду EQU. Наприклад:

```
Rozr    EQU        7
Rozr    MATRIX     ,3,5
```

У цьому прикладі ім'я Rozr матриці замінене на номер 7, за яким (MX7) потрібно звертатися до матриці. Якщо в команді MATRIX замість імені Rozr відразу записати число 7, то під час трансляції виникне зупинка із зазначенням помилки "Пропущена мітка оператора".

Перед початком моделювання або виконання команди CLEAR ON значення всіх елементів матриці мають дорівнювати нулю. Деяким або всім елементам за бажанням розробника можна присвоїти значення, які не дорівнюють нулю, або вони можуть бути переведені в невизначений стан командою INITIAL. Наприклад:

```
INITIAL  MX8(2,7),15.3
INITIAL  MX$Matr(3,8),25
INITIAL  MX$Mac(4,5),Kontr
INITIAL  Petit,3
INITIAL  Zsuv
INITIAL  Mist,UNSPECIFIED
```

У першому прикладі елементу матриці номер 8, який розташований на перетині другого рядка і сьомого стовпця, присвоюється початкове значення 15.3, у другому — елементу (3,8) матриці з іменем Matr — значення 25, у третьому — елементу (4,5) матриці з іменем Mac — значення змінної користувача Kontr. У четвертому прикладі замість операнда зазначено ім'я матриці Petit, тому всім елементам цієї матриці присвоюється значення 3. У п'ятому прикладі також замість операнда А стоїть ім'я матриці Zsuv, але операнд В не використовується, тому всім елементам матриці присвоюється значення 1. У шостому прикладі замість операнда В стоїть ключове слово UNSPECIFIED, тому всі елементи раніше створеної матриці Mist будуть переведені в невизначений стан. Цим переведенням матриця готується для використання в експерименті, в якому можуть бути пропущені, тобто не отримані під час експерименту дані. Невизначений стан елемента матриці засвідчує те, що даних немає. Якщо ж елементи матриці були б переведені в нуль, то це б сприймалось як отримання даних експерименту, які дорівнюють

нулю, що не відповідає дійсності.

Після експерименту матриця з його результатами передається бібліотечній процедурі дисперсійного аналізу (ANOVA), яка опрацьовує елементи, що перебувають у стані UNCSPECIFIED, як пропущені дані замість нульових значень.

Оператор MATRIX створює матрицю в поточній моделі. Матриця не може бути видалена з поточної моделі. Якщо оператор MATRIX вилучається з моделі, яка є у процесі виконання, то зв'язок з матрицею в цій моделі залишається. Матрицю можна перевизначити повторно іншим оператором MATRIX з тією ж міткою. Перед використанням матриця має бути визначена в операторі MATRIX. Матриця обмежена максимальним обсягом пам'яті, встановленим у журналі налаштувань моделі.

Для запису у процесі моделювання значень у матриці, а також для збільшення або зменшення значень, записаних у матриці, призначений блок MSAVEVALUE. Формат його запису такий:

MSAVEVALUE A,B,C,D

В операнді A задають ім'я або номер матриці. Крайнім правим символом операнда може бути знак + (режим накопичення) або знак — (режим віднімання). В операнді B задають номер рядка, в операнді C — номер стовпця матриці. Операнд D визначає значення, яке має зберігатися, додаватися або відніматися. Усі операнди є обов'язковими. Вони можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

MSAVEVALUE 4,5,P7,321

MSAVEVALUE X\$Matric-,3,7,V\$Serv

MSAVEVALUE Matr+,4,(Myst#5-3),M1

MSAVEVALUE (Kra+Q\$Cherga)+,(Roz-7),4,P\$Odn

MSAVEVALUE 3,Riad,Stovp,"Dovgyna"

У першому прикладі з входженням транзакта до блоку MSAVEVALUE значення елемента матриці номер 4, розташованого на перетині рядка 5 і стовпця, номер якого міститься в параметрі 7 цього транзакта, заміщається числом 321. У другому прикладі значення елемента на перетині

рядка 3 і стовпця 7 матриці, номер якої міститься у комірці зберігання *Matric*, зменшується на обчислене значення арифметичної змінної *Serv*.

У третьому прикладі номер стовпця знаходять як результат обчислення виразу в дужках (*Myst#5-3*), а потім значення визначеного елемента матриці *Matr* збільшується на величину відносного часу *M1* перебування транзакта в моделі.

У четвертому прикладі номер матриці і номер рядка елемента визначають як результати обчислення виразів у дужках (*Kra+Q\$Cherga*) і (*Roz-7*) відповідно. Після цього значення знайденого елемента збільшується на величину, яка міститься в параметрі з іменем *Odn* активного транзакта.

У п'ятому прикладі елементів матриці номер 3, розташованому на перетині рядка і стовпця, які визначають змінними користувача *Riad* і *Stovp*, присвоюється рядок "*Dovgyna*". Якщо змінним користувача раніше командою *EQU* не будуть присвоєні відповідні значення, то виникне зупинка внаслідок помилки.

Зупинка внаслідок помилки виникає також в тому випадку, коли із входом транзакта до блоку *MSAVEVALUE* не буде знайдена матриця з потрібним іменем або номером, або не буде знайдений елемент з відповідними номерами рядка та стовпця.

СЧА *MX* у блоці *MSAVEVALUE* при записі операнда *A*, як і СЧА *X* у блоці *SAVEVALUE* при записі операнда *A*, не використовується. Посилання на СЧА *MX* застосовується у виразах, функціях, операторі *INITIAL*.

З розглянутого формату команди опису матриці впливає, що матриця *GPSS* може мати до шести вимірів. Однак, як видно з формату блоку *MSAVEVALUE*, він дає змогу використовувати лише двовимірні матриці, тобто в ньому доступні лише перші два виміри. Решта індексів дорівнюють одиниці.

Якщо є необхідність використання матриць більшої розмірності, то потрібно створити одну або кілька *PLUS*-процедур. *PLUS*-процедури мають доступ до всіх елементів всіх матриць. Матриці, визначені командою *MATRIX*, є глобальними і доступні всім *PLUS*-процедурам. На час виконання *PLUS*-процедури можуть бути створені тимчасові матриці з локальною областю видимості.

LOGIC X A

Блок призначений для вмикання, вимикання або інвертування стану логічного ключа. Якщо логічний оператор *X* дорівнює *S* або *R*, то ключ *A*

встановлюється у стан "увімкнений" (Set) або "вимкнений" (Reset) відповідно. Якщо логічний оператор – I, то стан ключа інвертується. Блок LOGIC у режимі інвертування дозволяє чергувати альтернативи різного роду — маршрути, функції, числові значення й т. п.

13.2. Змінні та вирази

Алфавіт мови GPSS складається з латинських букв, цифр і спеціальних символів. Великі й малі букви розрізняються тільки в рядкових константах і коментарях. Рекомендується службові слова набирати великими літерами, а індивідуальні імена об'єктів — малими, починаючи із великої.

Арифметичні, умовні та логічні оператори

Обчислювальні вирази є комбінацією математичних операторів, бібліотечних функцій, СЧА і констант, що задовольняють правилам елементарної алгебри. Вони обчислюються згідно з ієрархією операторів в напрямку зліва направо. Порядок обчислень можна змінити за допомогою дужок.

Для позначення операцій, які виконують над елементами виразу, призначені арифметичні, умовні і логічні оператори. У GPSS World використовують такі оператори (у порядку убуття пріоритету):

- ^ — піднесення до степеня;
- # — множення; / — ділення; \ — ділення націло; @ — цілочисельний залишок;
- +, — — додавання й віднімання;
- >=, 'GE' — більше або дорівнює; <=, 'LE' — менше або дорівнює; >, 'G' — більше; <, 'L' — менше; =, 'E' — дорівнює;
- !=, 'NE' — не дорівнює;
- &, 'AND' — логічне множення;
- |, 'OR' — логічне додавання;
- 'NOT' — заперечення.

Якщо оператор потребує певного типу даних, то дані перетворюються автоматично. Наприклад, якщо над рядковою величиною виконується числова операція, то буде використаний числовий еквівалент цього рядка символів.

Оператор "#" використано для операції множення для того, щоб була можливість застосовувати оператор "*" для позначення непрямої

адресації. Проте користувач може змінювати значення операторів "#" і "*" (прапорець Switch * and #, в меню Edit — Settings — Simulation).

З математичних функцій дозволені ABS (абсолютне значення), ATN (арктангенс), COS (косинус), SIN (синус), TAN (тангенс), EXP (експонента), LOG (логарифм натуральний), SQR (корінь), INT (виділення цілої частини). Аргументи математичних функцій повинні бути взяті в дужки. Усі кути задаються в радіанній мірі.

Аргументи бібліотечних математичних функцій у всіх випадках автоматично перетворюються у числові значення. Допустимі значення аргументів — вираз. Обчислювані числові значення функцій — дійсного типу.

Змінні користувача

GPSS надає користувачеві можливість мати свої змінні для зберігання потрібних під час моделювання числових і рядкових даних. Змінні користувача створюються за допомогою команди EQU. Команда EQU створює іменований об'єкт (змінну користувача) і привласнює йому обумовлене виразом числове значення:

<ім'я змінної> EQU <вираз>

Приклади:

Myzminna	EQU	0.45
Rjad	EQU	15
Stovp	EQU	9
Name	EQU	"Name"

У наведених перших трьох прикладах змінним користувача Myzminna, Rjad, Stovp присвоюються числові значення, а змінній Name — рядок.

За допомогою змінних користувача імена об'єктів GPSS можна замінювати номерами. Для цього необхідно на початку моделі визначити змінні користувача командою EQU.

Імена, задані командою EQU, які в моделі використовують як мітки об'єктів, вважають мітками.

Змінні користувача можна застосовувати в PLUS-виразах, змінювати командами EQU або присвоюючими операторами в PLUS-процедурах. Якщо під час виконання експерименту з імітаційною моделлю передбачається дослідження впливу деяких чинників на шуканий показник, то чинники обов'язково мають бути змінними користувача.

Арифметичні змінні й арифметичні вирази

Арифметичні змінні дозволяють обчислювати арифметичні вирази, що складаються із стандартних числових атрибутів (СЧА). У виразі змінної використовують оператори, арифметичні дії і виклики бібліотечних функцій. Арифметична змінна є СЧА, який визначає користувач.

Якщо моделювання виконується не в режимі сумісності з GPSS/PC, то арифметична змінна в GPSS World може бути визначена командою VARIABLE або командою FVARIABLE, оскільки дія цих команд, а отже, й обчислювані значення, однакові. Формати команд:

<ім'я змінної> VARIABLE A

<ім'я змінної> FVARIABLE A

Операнд A – це вираз, який задає користувач.

Величиною арифметичної змінної є величина заданого користувачем арифметичного виразу. Значення арифметичної змінної можна використовувати як:

операнд; у цьому випадку значення арифметичної змінної може вказувати: номер об'єкта j; j логічного атрибута (блок GATE); номер параметра транзакта (блоки ASSIGN, INDEX, LOOP, MARK, SPLIT);

значення атрибута;

аргумент функції;

значення залежної змінної атрибутивної функції; аргумент таблиці;

операнд іншої арифметичної або булевої змінної. Приклади:

Bakir VARIABLE R\$Mybkr+S\$Mybkr

Kard VARIABLE Q\$Cherga-Q\$Bula

Dart VARIABLE P7+Q\$Mycher/3

Lasso VARIABLE N\$Tonic@12

У першому прикладі змінна Bakir є величиною, яка дорівнює сумі вільної ємності і поточного вмісту БКП Mybkr. Цей приклад показує, що розробник може задавати арифметичну змінну, величина якої є ємністю пристрою. У другому прикладі змінну Kard визначають як поточну довжину черги Cherga мінус поточна довжина черги Bula. У третьому прикладі змінну Dart обчислюють спочатку діленням поточної довжини черги Mycher на 3, а потім додаванням результату до значення параметра 7 активного транзакта. У четвертому прикладі Lasso — цілий залишок від ділення на 12 кількості входжень до блоку з іменем Tonic.

Значення арифметичного виразу обчислюється, коли активний транзакт потрапляє в блок, оператор якого серед своїх операндів містить посилання на арифметичні змінні. Підраховані значення є дійсними числами.

У GPSS World проміжні результати і значення СЧА не округлюються. Округлення може задавати лише сам користувач. Для цього можна використати операцію ділення націло, або бібліотечну функцію INT(A).

Наприклад:

```
Zminna1  VARIABLE      MX$Matr(P$Str,P$Sto)\Kor
```

```
Zminna1  VARIABLE      INT(MX$Matr(P$Str,P$Sto)/Kor)
```

Якщо процес моделювання виконується в режимі сумісності з GPSS/PC, то обчислення одного і того ж арифметичного виразу, заданого командами VARIABLE і FVARIABLE, здійснюється по-різному, оскільки в GPSS/PC значення всіх СЧА є цілого типу.

У режимі сумісності під час опису арифметичної змінної командою VARIABLE всі проміжні обчислення і кінцевий результат округлюються до цілого значення. Дійсні змінні (з плаваючою крапкою) визначає команда FVARIABLE. У цьому випадку округлюється до цілого значення лише кінцевий результат, тобто дійсні змінні є також цілочисельні.

Арифметична змінна і змінна з плаваючою крапкою в режимі сумісності з GPSS/PC не можуть мати однакових імен. Якщо вони мають однакові імена, то під час обчислення використовують останній з цих двох описів. Це відбувається тому, що СЧА V використовують як для звертання до арифметичних змінних, так і до змінних з плаваючою крапкою. Спосіб обчислення змінної V є оператором опису цієї змінної.

Розходження між результатами, отриманими під час обчислення змінних з плаваючою і фіксованою крапками, можна побачити з наведених нижче прикладів:

```
Obch1     FVARIABLE      10#(11/4)
```

```
Obch2     VARIABLE 10#(11/4)
```

Значення змінної Obch1 буде дорівнювати 27, оскільки число 10 потрібно помножити на 2.75 і від результату 27.5 взяти цілу частину. Змінна Obch2 дорівнює 20, оскільки результат проміжної операції 2.75 буде округлений до 2.

Оскільки GPSS World надає можливість використовувати вирази в операндах, то доцільно для спрощення моделей вираз, який неодноразово використовують у різних частинах моделі, визначити як змінну лише один раз, а потім посилатися на нього.

Булеві змінні

Булеві змінні дають можливість приймати рішення залежно від стану і значення багатьох об'єктів GPSS, використовуючи для цього лише один блок. У булевій змінній перевіряється одна або декілька логічних умов. Результат перевірки є 1, якщо задані умови задовольняються, і 0, якщо вони не задовольняються. Булевими змінними є логічні вирази, складені з різних стандартних числових атрибутів, у тому числі й інших булевих змінних.

Булеву змінну визначають командою B VARIABLE. Формат команди:

<ім'я> B VARIABLE A

Операнд A — логічний вираз. У виразі можна використовувати три типи операторів: логічні, булеві й оператори відношення. Кінцевий результат перетворюється в ціле значення 0, якщо дорівнює нулю, або в ціле значення 1, якщо відмінний від нуля.

Логічні оператори пов'язані з об'єктами апаратної категорії і використовуються для визначення стану цих об'єктів. Існують такі логічні оператори:

FVj дорівнює 1, якщо пристрій j доступний, інакше – 0;

FIj дорівнює 1, якщо пристрій j обслуговує переривання, інакше – 0; SFj

дорівнює 1, якщо БКП j заповнений повністю, інакше – 0;

SEj дорівнює 1, якщо БКП j порожній, інакше – 0; SVj

дорівнює 1, якщо БКП j доступний, інакше – 0;

LSj дорівнює 1, якщо логічний ключ j увімкнений, інакше – 0. Під j розуміють номер або ім'я об'єкта.

Приклади:

Stan1	B VARIABLE	FV\$Apparat1
Stan2	B VARIABLE	SF\$Apparat2
Stan3	B VARIABLE	SV\$Apparat3
Stan4	B VARIABLE	LS2

У першому прикладі булева змінна Stan1 дорівнює 1, якщо пристрій Apparat1 доступний, і – 0, якщо недоступний. У другому прикладі булева змінна Stan2 дорівнює 1, якщо БКП Apparat2 повністю заповнений. У

третьому прикладі булева змінна Stan3 дорівнює 1, якщо БКП Apparatz доступний. У четвертому прикладі булева змінна Stan4 дорівнює 1, якщо логічний ключ номер 2 увімкнений.

Оператори відношення здійснюють алгебраїчне порівняння операндів. Наприклад:

Vidn1	BVARIABLE	V\$Myzmin'G' 5
Vidn2	BVARIABLE	Q7'LE'P\$Porjadok
Vidn3	BVARIABLE	X\$Matr'GE'P8

Булева змінна Vidn1 дорівнює 1, якщо змінна Myzmin більше 5, інакше дорівнює 0. У другому прикладі булева змінна Vidn2 дорівнює 1, якщо поточна довжина черги номер 7 не перевищує параметра транзакта з ім'ям Porjadok. У третьому прикладі булева змінна Vidn3 дорівнює 1, якщо значення збереженої величини Matr не менше ніж значення параметра 8.

Булевих операторів є два: OR — АБО і AND — І. Оператор АБО перевіряє виконання хоча б однієї з умов, а оператор І потребує виконання обох умов. Наприклад:

Poriv1	BVARIABLE	FI\$Prib'OR'SF8
Poriv2	BVARIABLE	FI\$Prib'AND'SF\$Gos
Poriv3	BVARIABLE	(V2'G'5)'AND'(FV\$Prib1'OR'FV\$Prib2)

Булева змінна Poriv1 дорівнює 1, якщо виконується одна з умов: пристрій Prib обслуговує переривання або БКП номер 8 незаповнений. Булева змінна Poriv2 дорівнює 1, якщо виконуються обидві умови: пристрій Prib обслуговує переривання і БКП з іменем Gos незаповнений. У третьому прикладі змінна Poriv3 дорівнює 1, якщо виконуються обидві умови: значення змінної номер 2 більше ніж 5 і доступний один з пристроїв Prib1 або Prib2.

Дужки у третьому прикладі потрібні лише для задання певних булевих співвідношень. Дужки треба використовувати лише в тих випадках, коли вони необхідні.

Якщо булева змінна задається СЧА, як наприклад:

Myzmin	BVARIABLE	V\$Acc
--------	-----------	--------

то обчислюється значення арифметичної змінної на ім'я Acc, і якщо воно відмінне від нуля, то значення булевої змінної Myzmin буде дорівнювати 1, інакше — 0.

13.3. Зміна маршрутів транзактів

Блок GATE (впустити) в GPSS World використовується для визначення стану пристроїв, БКП і логічних ключів без зміни цього стану. Також блок GATE дозволяє змінювати маршрут руху транзакта за станом об'єкта, який перевіряється. Стан завжди має логічне значення. Формат блока:

GATE X A,B

Операнди:

X — вид умови, що перевіряється. При виконанні умови активний транзакт переходить до наступного блока;

A — ім'я або номер об'єкта, який перевіряється. Тип елемента повинен бути погоджений з умовою;

B — номер або ім'я блока, до якого направляється транзакт при невиконанні умови.

Умовний оператор X неявно задає тип об'єкта, який перевіряється. Допустимі значення:

для пристроїв: **U / NU** (зайнятий / вільний), **I / NI** (обслуговує переривання / неперерваний), **FV / FNV** (доступний / недоступний);

для БКП: **SE / SNE** (порожній / непорожній), **SF / SNF** (заповнений / незаповнений), **SV / SNV** (доступний / недоступний);

для логічних ключів: **LS / LR** (включений / виключений);

для транзактів: **M** — у блоці, заданому в полі A, у стані синхронізації перебуває транзакт, який належить до тієї самої сім'ї, що й транзакт, який перебуває у блоці GATE або робить спробу увійти до нього; **NM** — у блоці, заданому в полі A, у стані синхронізації немає жодного транзакта, який належить до тієї самої сім'ї, що й транзакт, який робить спробу увійти до блока GATE.

Операнди A і B можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Блок GATE працює у двох режимах:

відмови входження до блока;

дозволу входження та альтернативного виходу.

Якщо операнд B не використовується, GATE діє в режимі відмови. У випадку невиконання заданої умови транзакт переводиться до списку повторних спроб тестованого об'єкта. Коли стан об'єкта змінюється, заблокований транзакт знову активізується і повторюється перевірка

заданої блоком GATE умови. Якщо ця умова виконується, то транзакту дозволяється увійти до блока GATE і далі перейти до наступного за порядком блока.

У поєднанні з БКП GATE може використовуватися для формування пачок постійного обсягу, що дорівнює ємності БКП.

Для розгалуження маршруту на два напрямки використовуються двопозиційні логічні ключі: замок є/немає, червоний або зелений сигнал світлофора, табличка "Місць немає", "Перерва". У випадку триколірного світлофора використовують блок TEST і числову змінну (VARIABLE).

Приклади:

GATE NU Prystriy

GATE NU Pribor1,Mit

GATE SNF Pribor2,Mitka

У першому прикладі блок GATE не пропустить транзакта за умови зайнятості пристрою з іменем Prystriy. У другому прикладі у випадку зайнятості пристрою з іменем Pribor1 транзакт направляється до блока з міткою Mit. У третьому прикладі, якщо БКП з іменем Pribor2 незаповнений, тобто є вільні канали, то задана у блоці GATE умова виконується і транзакт буде направлений до наступного блока. Якщо БКП буде заповнений, то транзакт направляється до блока з міткою Mitka.

Блок TEST (перевірити) порівнює значення зазначених величин і управляє напрямком руху транзакта, ґрунтуючись на результаті порівняння.

TEST X A,B,C

Операнди:

X – вид умови, що перевіряється; A,

B – порівнювані величини;

C — номер або ім'я блока, до якого направляється транзакт при невиконанні умови.

Умовний оператор X може бути вираженням одним з шести умовних операторів: L – менше; LE – менше або дорівнює; E – дорівнює; NE – не дорівнює; G – більше; GE – більше або дорівнює.

Операнди A і B можуть бути іменем, числом, рядком, виразом у дужках, СЧА або СЧА*параметром.

Оскільки логічні значення в GPSS/W кодуються нулем й одиницею, у блоці TEST можна зіставляти й логічні значення.

Якщо операнд С зазначений, то транзакт завжди може увійти в блок TEST і залежно від співвідношення значень операндів А та В буде переданий або до наступного блока, або до блока, вказаного операндом С. Якщо поле операнда С порожнє, то транзакт у випадку невиконання умови не зможе увійти до блока TEST. Він переводиться до списку повторних спроб всіх об'єктів, які беруть участь у перевірці умови. Коли стан будь-якого з цих об'єктів змінюється, то транзакт зі списку повторних спроб активізується і перевірка здійснюється знову. Якщо умова виконується, то транзакт отримує дозвіл на вхід у блок TEST.

Блок TEST та булеві змінні можна використовувати для перевірки стану апаратних об'єктів.

У булевій змінній допускається використання таких логічних операторів:

F дорівнює 1, якщо пристрій зайнятий, інакше – 0;

SF дорівнює 1, якщо БКП заповнений повністю, інакше – 0; SE

дорівнює 1, якщо БКП порожній, інакше – 0;

SV дорівнює 1, якщо БКП доступний, інакше – 0, а

також СЧА пристроїв та БКП:

FC – кількість транзактів, які займали пристрій за допомогою блоків SEIZE і PREEMPT;

FR – коефіцієнт використання пристрою;

FT – середній час обслуговування пристроєм одного транзакта;

S – кількість зайнятих каналів БКП;

SA – середнє значення кількості зайнятих каналів БКП; SC – лічильник використання БКП;

SR – коефіцієнт використання БКП;

ST – середній час використання одного каналу БКП.

Отже, СЧА SE, SF і SV можна використовувати як у блоці GATE, так і в булевих змінних блока TEST.

Приклад використання булевої змінної та блока TEST для перевірки стану пристрою:

```
Bul  BVARIABLE    F$Server
```

```
TEST      E BV$Bul,0,Mitka
```

Послідовність проходження транзактами блоків у моделі також можна змінити за допомогою блоків TRANSFER, DISPLACE та LOOP.

Блок TRANSFER (передати) змінює траєкторію руху активного транзакта. Формат блока:

TRANSFER A,B,C,D

Операнди:

A — режим блока, обирається з множини {BOTH, ALL, PICK, FN, P, SBR, SIM}, може бути також дробовим числом, ім'ям, константою, СЧА;

B — номер або ім'я блока (у режимі P — ім'я параметра);

C — номер або ім'я блока (у режимі FN або P — збільшення);

D — збільшення номера блока для режиму ALL. За замовчуванням дорівнює 1.

Усі режими блока TRANSFER, крім безумовного, вибіркового, тобто відрізняються один від одного способом вибору чергового блока, до якого має бути направлений активний транзакт. Операнд A задає цей режим вибору. Блок TRANSFER може працювати в дев'яти режимах:

, (за замовчуванням) — безумовний;

. — статистичний, тобто вибір випадковим чином одного з двох блоків;

BOTH — послідовний вибір одного з двох блоків;

ALL — послідовний вибір одного з кількох блоків;

PICK — вибір випадковим чином одного з кількох блоків;

FN — функціональний;

P — параметричний;

SBR — підпрограмний;

SIM — одночасний.

Операнд A, крім перерахованих значень, також може бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром.

Операнди B і C визначають можливі імена наступних блоків або їхнє розташування. Вони також можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Використання цих значень розглянемо під час ознайомлення з конкретними режимами роботи. Якщо операнд B відсутній, то планувальник записує замість нього ім'я або номер блока, який розташований після блока TRANSFER.

Розглянемо режими роботи блока TRANSFER.

Режим безумовної передачі. У цьому режимі операнд A не використовують, а операнд B зазначає ім'я блока, до якого направляється транзакт. Блок TRANSFER завжди відкритий для входження транзактів. Наприклад:

TRANSFER ,Mit1

Потрапивши у блок TRANSFER, транзакт відразу намагається увійти у блок з міткою Mit1. Якщо транзакт отримує відмову на входження, то він залишається у блоці TRANSFER.

Статистичний режим. Значення операнда А визначає імовірність, з якою транзакт направляється до блока, зазначеного в полі операнда С. З імовірністю $1-A$ транзакт направляється до блока, вказаного в полі операнда В (до наступного блока, якщо поле В порожнє).

Імовірність у полі А може бути задана безпосередньо десятковим дробом, який починається з крапки. Наприклад блок

TRANSFER .75,Tut,Tam

з імовірністю 0,75 направляє транзакти до блока з іменем Tam, а з імовірністю 0,25 — до блока з іменем Tut.

Якщо поле А починається не з десяткового дробу і не містить одного з ключових слів — ознак інших режимів роботи, то його значення розглядається як кількість тисячних часток імовірності передачі. Наприклад, попередній блок TRANSFER можна записати також у такому вигляді:

TRANSFER 750,Tut,Tam

Числове значення операнда А можна задавати будь-яким СЧА. Можливі такі випадки:

- 1) значення операнда А менше або дорівнює нулю;
- 2) значення операнда А дорівнює або більше ніж 1000;
- 3) значення операнда А більше від нуля, але менше ніж 1000. Якщо

підраховане значення операнда А менше або дорівнює нулю, то буде здійснюватися безумовне направлення транзакта до блока В. Якщо значення операнда А більше або дорівнює 1000, то буде здійснюватися безумовна передача транзакта до блока С. У третьому випадку блок TRANSFER працює у звичайному режимі.

Наприклад, запис

TRANSFER P5,Tam

означає, що тризначне число, записане в параметрі 5 транзакта, який потрапив у блок TRANSFER, інтерпретується як імовірність (в частках тисячі) того, що транзакт передаватиметься до блока Tam, а в решті випадків — до наступного блока, оскільки операнд В не використовується. Режим статистичної передачі зручно використовувати для розгалуження маршруту руху транзактів на два напрямки.

Можна визначити генератор — джерело випадкових чисел для блока TRANSFER. Для цього, вибравши в меню Edit — Settings закладку Random Numbers, треба ввести номер генератора в полі блока TRANSFER.

Режим BOTH. Якщо в полі операнда А використовується ключове слово BOTH (обидва), то блок TRANSFER працює в режимі BOTH. У цьому режимі кожний транзакт, увійшовши в блок TRANSFER, перевіряє дві можливості. Спочатку він намагається увійти до блока В (або до наступного блока, якщо поле В порожнє), а якщо це не вдається, тобто блок В не дозволяє входження, то — до блока С. Якщо й ця спроба невдала, то транзакт затримується у блоці TRANSFER до такої зміни умов у моделі, які б дозволили увійти до одного з блоків В або С. Блоку В надається перевага, якщо така можливість виникне одночасно.

Приклад:

```
.....  
..... TRANSFER  
BOTH,,Mitka  
SEIZE Prist1  
ADVANCE 15,5  
RELEASE Prist1  
.....  
..... Mitka SEIZE  
Prist2  
ADVANCE 20  
RELEASE Prist2  
.....
```

Транзакти потрапляють у блок TRANSFER, який працює в режимі BOTH. Якщо в момент надходження транзакта пристрій Prist1 може його прийняти, то блок TRANSFER направить транзакт до наступного блока, тобто до блока SEIZE. Якщо ж у момент надходження транзакта пристрій не може його прийняти (пристрій зайнятий, обслуговує переривання чи перебуває у стані неготовності), то транзакт буде направлений до блока SEIZE з іменем Mitka. Якщо жоден блок не зможе прийняти транзакт, то він залишається у блоці TRANSFER доти, поки якийсь пристрій не прийме його.

При моделюванні замість блока TRANSFER, який працює в режимі BOTH, можна використати також блок GATE або TEST, замінивши рядок програми

```
TRANSFER BOTH,,Mitka
```

на такий:

GATE NU Prist1,Mitka або

на такий:

TEST E F\$Prist1,0,Mitka

У цьому варіанті моделі транзакт проходить до блока SEIZE, якщо пристрій вільний (перевірка проводиться відповідно блоком GATE або TEST).

Режим ALL. Блок TRANSFER у режимі ALL, як і в режимі BOTH, виконує роль "диспетчера", але на відміну від блока BOTH, він розподіляє транзакти між кількома пристроями.

Для роботи в режимі ALL необхідно в полі операнда А використати ключове слово ALL. У цьому режимі, увійшовши у блок TRANSFER, транзакт перевіряє можливість входження до будь-якого блока, починаючи з блока, зазначеного операндом В, і закінчуючи блоком, вказаним операндом С. Операнд D визначає крок d зміни номера блока, який перевіряється. Значення кроку дає змогу опитувати певні блоки, розташовані між тими, які задані операндами В і С.

Спочатку транзакт, увійшовши у блок TRANSFER, намагається увійти до блока В. Якщо блок В зайнятий, то транзакт намагається увійти до блоків з номерами $v+d$, $v+2d$, ..., $x=v+nd$, де v — номер блока В, x — номер блока С, n — деяке натуральне число.

Якщо операнд С не використовують, то перевіряється лише один блок. Оскільки, зазвичай, у полях операндів В і С записують мітки блоків, то блоки потрібно розташовувати так, щоб під час присвоєння номера різниця між номерами блоків, зазначених операндами В і С, була кратною крокові, вказаному в полі операнда D. Наприклад:

TRANSFER ALL,Mit1,Mit2,4

Тут режим ALL допустимий, якщо різниця між номерами, присвоєними блокам Mit1 і Mit2, кратна 4. У моделі це може виглядати так.

Приклад. Замовлення розподіляють між трьома пристроями. Чергове замовлення обслуговується тим пристроєм, який першим звільнився.

GENERATE 10,5 TRANSFER

ALL,Mit1,Mit2,4

;Імітування роботи пристрою 1 Mit1

SEIZE Prib1

```
ADVANCE 10  
RELEASE Prib1  
TERMINATE
```

;Імітування роботи пристрою 2 SEIZE

```
Prib2  
ADVANCE 15  
RELEASE Prib2  
TERMINATE
```

;Імітування роботи пристрою 3 Mit2

```
SEIZE Prib3  
ADVANCE 20,10  
RELEASE Prib3  
TERMINATE
```

;Таймер

```
GENERATE 480  
TERMINATE 1
```

Після надходження транзакта у блок TRANSFER послідовно перевіряється можливість входження спочатку до блока SEIZE, позначеного міткою Mit1. Потім, якщо цей блок не дозволяє входження, до другого блока SEIZE без мітки, оскільки різниця між номерами цього блока та блока з міткою Mit1 дорівнює 4. Якщо входження до другого блока SEIZE неможливе, то перевіряється третій блок SEIZE з міткою Mit2. Якщо й він зайнятий, то транзакт залишається у блоці TRANSFER, і з кожною зміною поточного модельного часу перевірки можливості входження знову повторюються спочатку, тобто з блока SEIZE з міткою Mit1.

У блоці TRANSFER операнди B і C можуть бути не мітками, а номерами блоків. Зокрема, якщо в моделі попереднього прикладу записати

```
TRANSFER ALL,3,11,4
```

то результат моделювання буде такий самий (нумерація починається з блока GENERATE).

Зауважимо, що у режимі ALL (як і у режимі BOTH) у випадку, коли можливий перехід до декількох блоків, то блоки з меншими номерами мають деяку перевагу перед блоками з більшими номерами.

Режим PICK вибирає подальше призначення транзакта в інтервалі від В до С (послідовні номери) з рівними ймовірностями незалежно від можливого блокування.

Для роботи блока TRANSFER у режимі PICK необхідно в полі операнда А використати ключове слово PICK. У цьому режимі з послідовності блоків з номерами $v, v+1, v+2, \dots, n$, де v — номер блока, зазначеного операндом В, а n — номер блока, вказаного операндом С, випадково вибирається номер одного блока, до якого має перейти транзакт. Усі блоки, враховуючи й вказані операндами В і С, вибираються з однаковою ймовірністю $1/(n-v+1)$. Транзакт намагається перейти лише до вибраного для нього блока. Якщо транзакт не може відразу перейти до наступного блока, то він чекатиме у блоці TRANSFER до того часу, поки не буде знята умова блокування. Номер блока С має бути більший або дорівнювати $v+1$.

Приклад. Замовлення розподіляються між чотирма пристроями, які мають однаковий час обслуговування. Ймовірність потрапляння до будь-якого з пристроїв $P=1/4=0,25$. Якщо в режимах BOTH і ALL чергове замовлення займало той пристрій, який раніше звільнявся, то в режимі PICK замовлення чекає на звільнення того пристрою, на який воно було розподілене. Оскільки блоки, до яких передаються замовлення, в моделі розташовані не послідовно, то використовується послідовність блоків TRANSFER у режимі безумовної передачі.

```
GENERATE                10,5
TRANSFER PICK,Mit1,Mit2
Mit1 TRANSFER ,Mit3
      TRANSFER ,Mit4
      TRANSFER ,Mit5
Mit2 TRANSFER ,Mit6
;Імітування роботи пристрою 1
Mit3 SEIZE Prib1
      ADVANCE 10,5
      RELEASE Prib1
      TERMINATE
;Імітування роботи пристрою 2
Mit4 SEIZE Prib2
      ADVANCE 10,5
      RELEASE Prib2
```

TERMINATE

;Імітування роботи пристрою 3

Mit5 SEIZE Prib3

ADVANCE 10,5

RELEASE Prib3

TERMINATE

;Імітування роботи пристрою 4

Mit6 SEIZE Prib4

ADVANCE 10,5

RELEASE Prib4

TERMINATE

;Таймер

GENERATE 480

TERMINATE 1

Режим функції діє, коли операнд А є FN. Під час входження транзакта у блок TRANSFER підраховується значення функції, ім'я якої задане операндом В. Якщо результат не цілий, то від нього береться ціла частина. Для визначення номера наступного блока отримане значення додається до значення операнда С (значення останнього може бути записане нулем). Якщо блок з обчисленим номером зайнятий, то транзакт залишається у блоці TRANSFER до того часу, поки не зможе перейти саме до цього блока. Наприклад:

TRANSFER FN,Vybir,P4

TRANSFER FN,8,P4

У першому прикладі транзакт направляється до блока, номер якого планувальник визначає як суму цілої частини обчисленого значення функції з іменем Vybir і значення параметра 4 цього транзакта. У другому прикладі все так само, лише функція задана не іменем, а номером.

Режим параметра. Для задання параметричного режиму потрібно в полі операнда А вказати ключове слово Р. У цьому режимі активний транзакт направляється до блока, номер якого визначається як сума значення параметра та значення операнда С. Якщо операнд С не використовують, то значення параметра приймається за номер нового блока. Наприклад:

TRANSFER P,Param,5

TRANSFER P,4,5

У першому прикладі транзакт направляється до блока, номер якого дорівнює сумі значення параметра з іменем Param і значення операнда C, тобто 5. У другому прикладі все так само, лише параметр заданий не іменем, а номером.

Режим підпрограми. Для задання підпрограмного режиму потрібно в полі операнда A зазначити ключове слово SBR. У цьому режимі активний транзакт завжди направляється до блока, номер якого вказаний операндом B. Номер блока TRANSFER заноситься в параметр, вказаний операндом C. Наприклад:

TRANSFER SBR,Mitka,Par

Увійшовши у блок, транзакт переходить до блока з міткою Mitka, а номер блока TRANSFER записується в параметр з іменем Par. Якщо такого параметра немає, то планувальник його створює.

Для повернення з підпрограми використовують блок TRANSFER у параметричному режимі.

Ця технологія дозволяє організувати перехід транзакта з поверненням, аналогічний передачі керування на підпрограму з поверненням на продовження основної програми.

Режим SIM (одночасний) використовують у випадку, коли потрібне одночасне виконання кількох умов. Кожен транзакт має свій індикатор затримки SIM. У цьому індикаторі записується результат кожної спроби входження транзакта до наступного блока. Якщо планувальник виявляє умови, які перешкоджають входженню транзакта до блока, то індикатор SIM цього транзакта дорівнює 1. Якщо всі умови переходу до наступного блока задовольняються, то індикатор SIM дорівнює нулю. Якщо не виконується хоча б одна з умов, то індикатор SIM дорівнює 1. У таких випадках операнд C вказує блок, у якому перевірялась перша умова, і транзакт виконує перевірки всіх умов, поки вони не будуть задовольнятися одночасно. Блок ADVANCE також встановлює індикатор SIM, який дорівнює нулю. Перевірка станів пов'язана з блоками, які можуть затримувати транзакти. Для перевірки умов використовують блок GATE.

Приклад:

Mit GATE SE Bkp1
GATE SE Bkp2
GATE SE Bkp3
TRANSFER SIM,,Mit
ENTER Bkp4

У наведеному прикладі транзакт не може перейти до блока ENTER, якщо БКП Bkp1, Bkp2 і Bkp3 одночасно порожні. Під час входження транзакта у блок TRANSFER перевіряється значення його індикатора SIM. Якщо транзакт був затриманий в якомусь з блоків GATE, то його індикатор дорівнює 1, і під час перевірки індикатора у блоці TRANSFER виявляється, що транзакт був затриманий. Тому він відсилається до того блока, де здійснювалася перша перевірка (Mit), і вся послідовність перевірок повторюється.

Під час затримки транзактів у блоках ASSEMBLE, GATHER або MATCH індикатор SIM не дорівнює 1.

Режим SIM використовують зрідка, оскільки за допомогою булевих змінних і блока TEST можна ефективніше організувати управління станом великої кількості об'єктів.

Блок LOOP (зациклити) призначений для організації в моделі циклів. Він модифікує параметр й управляє рухом активного транзакта, виходячи з результату цієї модифікації. Формат блока:

LOOP A,B

Операнди:

A — параметр транзакта або параметр циклу, в якому міститься число – кількість повторень якого-небудь сегмента моделі;

B – мітка блока, з якого починається цикл.

Обидва операнди можуть бути іменем, додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

LOOP KilPovt,Pochat

Нехай блок з міткою Pochat є початком циклу, тобто розташований раніше від блока LOOP. Коли транзакт, пройшовши ділянку моделі, яка починається блоком з міткою Pochat, увійде до блока LOOP, значення його параметра з іменем KilPovt зменшується на 1. Якщо це значення не дорівнює нулю, то транзакт переходить до блока з міткою Pochat, тобто цикл повторюється. Якщо ж після віднімання 1 значення параметра циклу дорівнює нулю, отже, виконана задана кількість повторень, то транзакт переходить до наступного блока.

Блок LOOP можна використовувати для моделювання СМО з відмовами і заданою кількістю повторних спроб.

DISPLACE A,B,C,D

Блок DISPLACE призначений для відшукування будь-якого транзакта і переміщення його до нового блока зі збереженням інтервалу модельного часу. Блок має такий формат:

A – номер транзакта, який потрібно перемістити;

B — мітка блока, до якого передається транзакт, вказаний операндом A;

C — номер або ім'я параметра цього транзакта, до якого записується час, що залишився до завершення його обслуговування, якщо транзакт перебував у списку майбутніх подій;

D – мітка альтернативного блока для активного транзакта.

Усі операнди можуть бути іменем додатним цілим числом, виразом у дужках, СЧА або СЧА*параметром. Наприклад:

DISPLACE (P6+7),Mit1,Result,Mit2

Тут операнд A є виразом у дужках. Цей вираз обчислюється та округлюється до цілого. Отриманий результат є номером транзакта, який треба перемістити. Далі блок DISPLACE розшукує цей транзакт. Для останнього можливі випадки:

транзакт є в моделі і не перебуває у списку майбутніх подій;

транзакт є в моделі і перебуває у списку майбутніх подій;

транзакта із зазначеним номером немає в моделі.

У першому випадку транзакт передається до блока з іменем Mit1. У другому випадку визначається час, який залишився до його повернення у процес моделювання, і записується в параметр з іменем Result. Якщо параметра з таким іменем немає, то він автоматично створюється. Транзакт також передається до блока з міткою Mit1. У третьому випадку, тобто коли в моделі немає транзакта з потрібним номером, активний транзакт, який увійшов до блока DISPLACE, скеровується до блока з міткою Mit2. Якщо операнда D немає, то активний транзакт переходить до наступного блока.

Коли транзакт переходить до нового блока, він вилучається зі списків:

майбутніх подій;

відкладених переривань (для транзактів, які здійснюють переривання);

затримки (у порядку пріоритету);

користувача;

повторних спроб.

У той же час він не вилучається зі списків:

поточних подій;

переривань (для перерваних транзактів); груп.

Під час пересування перервані виконання у пристроях не припиняються. Це означає, що транзакт продовжує займати пристрій.

Висновки

1. Зв'язок між транзактами реалізується через збережувані величини, які бувають скалярними і матричними. Значення параметрів транзактів і збережуваних змінюються лише за прямою вказівкою користувача. Перед початком моделювання за замовчуванням вони дорівнюють нулю.
2. Матриці створюються у поточній моделі й описуються оператором MATRIX. Матриця обмежена максимальним обсягом пам'яті, встановленим у журналі налаштувань моделі. Матриця не може бути видалена з поточної моделі, її можна перевизначити повторно іншим оператором MATRIX з тією ж міткою.
3. Для проведення розрахунків (арифметичних, перевірки умов) призначені обчислювальні вирази, які є комбінацією математичних операторів, бібліотечних функцій, СЧА і констант, що задовольняють правилам елементарної алгебри.
4. Для зберігання потрібних під час моделювання числових і рядкових даних використовуються змінні.
5. Булеві змінні дають можливість приймати рішення залежно від стану і значення об'єктів GPSS.
6. Блоки GATE, TEST, TRANSFER, LOOP і DISPLACE дозволяють змінити траєкторію руху транзакта по моделі.
7. Блоки GATE, TEST і TRANSFER використовують для моделювання різних СМО з втратами замовлень.

Контрольні запитання та завдання

1. Дайте визначення збережуваної величини і матриці в GPSS World. Як і для чого вони використовуються в моделях?
2. Назвіть особливості роботи з матрицями в GPSS World.

3. Дайте визначення змінної та виразу в GPSS World. Як і для чого вони використовуються в моделях?
4. Яким чином зорганізуються посилання на збереження величини, матриці, комірки матриць, змінні.
5. Назвіть блоки, що змінюють маршрути транзактів у моделі. Вкажіть особливості їх роботи.
6. Назвіть та опишіть режими роботи блока TRANSFER.