

ПСЕВДОЯЗЫК ОПИСАНИЯ СЦЕПЛЕННЫХ ПРОЦЕССОВ (ПОСП)

Общие положения

Идентификатор - последовательность букв, цифр и символов. В идентификаторах разрешается использование букв и символов любых алфавитов. Типы идентификаторов:

- а) *простой* – последовательность букв и цифр, начинающаяся с буквы;
- б) *составной* – последовательность простых идентификаторов, соединенных знаком подчеркивания (_);
- в) *стандартный* – фиксированный простой или составной идентификатор.

Запись описаний и операторов языка сопровождается служебными словами, которые пишутся русскими буквами и подчеркиваются.

Запись чисел соответствует общепринятым в математике правилам.

Любой оператор может быть помечен меткой. *Метка* – идентификатор, отделяется двоеточием от оператора.

Список – линейная последовательность элементов (идентификаторов), разделённых запятой.

Строка символов – любая последовательность символов, кроме кавычек, помещенная в кавычки.

Блоки типа *агрегат* и *контроллер* содержат единственный инициатор по умолчанию. Начальное местонахождение инициатора указывается в блоке в разделе “Описание”.

Стандартные идентификаторы

ВРЕМЯ – переменная типа скаляр, ее значение - текущее время в модели.

RAND – стандартная функция. Вычисляет очередное значение псевдослучайной переменной RAND в [0,1] с равномерным законом распределения.

ИНИЦИАТОР – переменная типа ссылки, принимающая значение ссылки на локальную среду текущего инициатора.

В выражениях могут использоваться знаки и символы любых операций из области математики, лингвистики и пр. В выражениях в качестве стандартных идентификаторов используются имена функций, например, SIN, COS, EXP, SQRT и т.д. Перечень таких идентификаторов может быть задан пользователем для каждой конкретной программы.

Объекты языка

Объектами языка являются: *простая переменная; переменная; блок; инициатор.*

Каждый объект имеет тип, имя и значение.

Имя объекта есть идентификатор. *Значение объекта* есть его содержание.

Тип простой переменной: скаляр; ссылка; метка.

Тип переменной: скаляр; ссылка; метка; вектор; пространство.

Тип блока: агрегат; процессор; контроллер.

Тип инициатора: ссылка.

Описание типов

Скаляр – одно неделимое значение (число, логическое значение либо строка символов).

Синтаксис: <список имен простых переменных> - скаляр(ы)

Ссылка – простая переменная типа скаляра, значением которой является адрес объекта.

Синтаксис: <список имен простых переменных> - ссылка(и)

Метка – простая переменная типа скаляра, значением которой является адрес помеченного ею оператора.

Синтаксис: <список имен простых переменных> - метка(и)

Вектор – линейно - упорядоченная совокупность скаляров либо векторов (определение рекурсивно).

При описании вектора задаются границы индексов и типы элементов. Если левая и правая границы совпадают, то указание правой границы может быть опущено.

Пространство – n-мерное пространство однотипных переменных. Описание пространства задает размерность пространства и границы индексов по каждому измерению.

Блок описывается как класс (ООП). Синтаксис:

блок - <тип блока> <имя блока>

описание <список групп описаний>

все описание;

алгоритм

<программа>

все алгоритм;

все блок.

< тип блока > ::= агрегат | процессор |
контроллер | параметры

Список групп описаний содержит группы описания параметров с указанием: внешние, внутренние, типов, внешних блоков, меток, необходимых комментариев и договоренностей.

Программа – последовательность операторов ПОСП. Все описания и операторы в программе отделяются друг от друга символом < ; >. Комментарии могут вводиться в любом месте программы и отделяться от операторов двойным слешем < // >.

Операторы (стандартные операции)

Арифметическое (логическое) выражение есть последовательность арифметических (логических) операций над параметрами, имеющая целью получить некоторое конкретное числовое (логическое) значение.

Над параметрами могут выполняться любые арифметические и логические операции.

Полученное значение присваивается параметру с помощью **операции присваивания**: <переменная> := <выражение>

Операции над инициатором

Безусловное направление инициатора

Синтаксис: направить инициатор на <метка> [блок <имя блока>]

Последняя часть может быть пропущена, если эта операция происходит в одном и том же блоке.

Условное направление инициатора

Синтаксис: если <логическое выражение> то направить инициатор на <метка> [иначе на <метка>]

Последняя часть (*иначе*) может быть пропущена, если инициатор продолжает движение по программе.

Векторная форма условного направления инициатора

Синтаксис:

если

V1 направить инициатор на <метка>

:

VN направить инициатор на <метка>

иначе направить на <метка>

Здесь V1... VN - логические выражения.

Оператор задержки инициатора

Синтаксис: ждать <логическое выражение>

Оператор задерживает инициатор до выполнения логического условия. Условие может содержать переменную ВРЕМЯ. Оператор выполняет функции условного элементарного оператора, совмещая временной, логический и смешанный виды.

Векторная форма оператора задержки инициатора

Синтаксис:

ждать

V1 направить инициатор на <метка>

:

VN направить инициатор на <метка>

Здесь V1...VN - логические выражения.

Активизация инициатора

Синтаксис:

активизировать инициатор из <имя простой переменной типа ссылки > в блок < имя блока> на метку <метка>

Оператор активизирует параметр типа ссылки, превращая его в инициатор и направляя на сцепление с помеченным оператором указанного блока.

Пассивизация инициатора

Синтаксис: пассивизировать инициатор в параметр <имя параметра>

Этот оператор выполняет действие, обратное оператору активизировать, лишая текущий инициатор свойства инициализации и направляя ссылку на локальную среду из инициатора в указанный параметр ссылочного типа.

Операции над объектами

Создание объекта

Синтаксис: создать <имя объекта> типа <тип>

Операция создает объект указанного типа и вводит его в программу.

Уничтожение объекта

Синтаксис: уничтожить <тип объекта> <имя объекта>

Операция уничтожает объект с заданным именем и выводит его из программы.

Операции над ссылочными переменными

Присваивание значения ссылке

Синтаксис: <имя ссылки> := ссылка на [тип объекта] <имя объекта>

Ссылке присваивается адрес объекта.

Разыменование ссылки

Синтаксис: <имя ссылки> → <тип переменной>

Операция позволяет по ссылке определять значение переменной указанного типа. *Значение переменной* есть результат операции. Если операция встречается в выражении, то вышеуказанная запись заключается в скобки ().