Naítanie hodnôt a zoznamov vektorov

Do výpotu vektora najastejšie vstupuje iný vektor (môže by taktiež vypoítaný), alebo zoznam (pole) vektorov. Poet vektorov vstupujúcich do výpotu nie je umelo obmedzený, avšak ím viac vektorov je potrebné spracova, tým viac stúpa asová a pamäová náronos výpotu.

Pri naítaní hodnôt a zoznamov vektorov sú rozlišované dva typy funkcií. Prvý typ sprístupuje obsah entít cez ich meno a druhý vracia naítaný obsah do lokálnej premennej príslušného typu. Odporúa sa použi druhý typ - naítanie obsahu do lokálnej premennej.

- Sprístupnenie obsahu:
  - %LoadGroup
  - %LoadScenario
  - %LoadVector
- Naítanie do lokálnej premennej:
  - %LoadGroupData
  - %LoadScenarioData
  - %LoadVectorData

Nasledovný príklad naítava hodnoty vektora *Vector.Src* v rovnakom asovom rozsahu, ako je naítavaný výsledný vypoítaný vektor s rovnakým krokom ako má výsledný vypoítaný vektor.

Naítanie hodnôt vektora

```
VECTOR _v;

-- Hodnoty budú prístupné cez identifikátor Vector.Src
%LoadVector(Vector.Src, _BT, _ET, _STEP);

-- Hodnoty budú prístupné cez premennú _v
_v := %LoadVectorData("Vector.Src", _BT, _ET, _STEP);
```

Naítanie vektora s parametrami

Do výpotu vypoítaného vektora je možné volitene podsunú parametre. Poet parametrov odovzdaných do výpotu vektora je možné získa funkciou %GetParamsNr. Parameter je identifikovaný svojim indexom. Hodnotu kontrétneho parametra je možné priradi do lokálnej premennej funkciou %GetParamValue. Ak hodnota parametra nie je kompatibilná s typom lokálnej premennej, do ktorej je priraovaná, je generovaná chyba.

Vector.Calc

```
REAL _value;
VECTOR _v;

IF %GetParamsNr() > 0 THEN
  -- Priradí hodnotu parametra na indexe 1 do premennej _value
  %GetParamValue(1, _value);
END IF;
%CreateVectorInit(_v, _value, _BT, _ET, _STEP);

RETURN _v;
```

Naíta vektor *Vector.Calc* s podsunutím hodnoty prvého parametra 2,5.

Vector.Result

```
%LoadVektorWithParams(Vector.Calc, _BT, _ET, _STEP, 2.5);
RETURN Vector.Calc;
```

Pre zvýšenie prehadnosti kódu a zjednodušenie práce s parametrami je odporúané použitie pomenovaných parametrov.

Naítanie vektora s pomenovanými parametrami

Ide o naítanie vektora, ktorý má jasne menom definované parametre. Hodnoty sú týmto parametrom automaticky pridelené pri naítaní vektora. Nezadané parametre budú obsahova neplatnú hodnotu. Nasledovný príklad pozostáva z dvoch vektorov. Vektor, ktorý spracováva výpoet s parametrami *(Vector. Calc)* a vektor, ktorý naítava vektor *Vector.Calc* a odovzdáva mu hodnoty parametrov *(Vector.Result).*

Nasledujúci predpis zobrazuje vektor Vector.Calc, ktorý má definovaný pomenovaný parameter _value typu reálne íslo (1).

Vector.Calc

```
PARAM REAL _value; -- (1)
VECTOR _v;

%CreateVectorInit(_v, _value, _BT, _ET, _STEP);

RETURN _v;
```

Vektor *Vector.Result* už len volá naítanie vektora *Vector.Calc*, ktorému odovzdá hodnotu parametra *_value* (1). Ako hodnotu parametra je možné zada akýkovek výraz, ktorého výsledný typ je kompatibilný s typom parametra.

Vector.Result

```
VECTOR _v;

v := %LoadVectorData("Vector.Calc", _BT, _ET, _STEP, , , _value => 2.5); -- (1)

RETURN _v;
```

Naítanie zoznamu vektorov

Naítanie zoznamu vektorov je zvyajne realizované naítaním scenára vektorov (%LoadScenario, %LoadScenarioData) alebo skupiny vektorov (%LoadGroup, %LoadGroupData). Je však možné naíta aj ubovolný zoznam vektorov na základe SQL príkazu (%LoadFilter, %LoadFilterSimple). Ak do výpotu vstupujú aj hodnoty vektorov z naítaného zoznamu, je potrebné ich naíta/sprístupni.

Naítanie zoznamu vektorov

```
INT _i, _nr;
VECTOR _v;

%CreateVector(_v, _BT, _ET, _STEP);
%LoadScenario(Scenario.Test, 1);
_nr := %ValuesNr(Scenario.Test);
IF _nr > 0 THEN
  -- Sprístupní hodnoty vektorov v zozname
  FOR _i = 1 TO _nr LOOP
    %LoadVector(Scenario.Test[_i], _BT, _ET, _STEP);
  END LOOP;

  -- Vypoíta súet hodnôt na rovnakých indexoch
  _v := %Sum(Scenario.Test);
END IF;
```

Štruktúra VECTORS môže okrem EDA vektorov obsahova aj odkazy na lokálne vektory (premenné typu VECTOR). V prípade odovzdania takejto štruktúry do iného vypoítaného vektora, budú hodnoty zdrojového lokálneho vektora dostupné na modifikáciu aj vo vnorenom vypoítanom vektore, priom zmena hodnôt sa prejaví v každej úrovni vnorenia. Nasledovný predpis predstavuje jednoduchý príklad použitia odkazov na lokálne vektory v poli vektorov.

```
VECTOR _v1, _v2;
VECTORS _w;
INT _i, _j;

%CreateVector(_v1, 5);
%CreateVector(_v2, 10);

-- Do poa vektorov pridá lokálne vektory _v1 a _v2
_w := _w + _v1 + _v2;

-- Modifikuje hodnoty lokálnych vektorov nepriamo cez zoznam vektorov
FOR _i = 1 TO %ValuesNr(_w) LOOP
 FOR _j = 1 TO %ValuesNr(_w[_i]) LOOP
  _w[_i][_j] := _i + _j;
 END LOOP;
END LOOP;
```

Súvisiace stránky:

Vytvorenie vektora
%CreateVectorInit
%ValuesNr