

Beanshell scriptelés

Megjegyzendő, hogy a Beanshell scriptelés tapasztalt felhasználók számára ajánlott, mivel scriptek írásához szükség van programozói ismeretekre.

Részletesebb Beanshell kézikönyv a Beanshell honlapján található az alábbi címen: <http://www.beanshell.org/>. Ez az útmutató csak a *View Creation Wizard*-ban írandó scriptekhez hasznos részleteket tartalmazza.

A scriptek több sorosak is lehetnek. Visszatérési értékük mindig vagy az utolsó kiértékelt kifejezés vagy metódus hívás eredménye, vagy az az érték, amit a `return` utasítással adunk meg.

A MEME-ben két fajta script létezik: skalár, vagy tömb visszatérési értékű. Mivel a MEME minden paraméterből létrehoz egy globális változót, a scriptekben a paraméterekre a nevükkel lehet hivatkozni. Skalár visszatérési értékű script esetén a hivatkozott paraméter típusa egyetlen érték (`double`, `int`, `String`, stb.), míg a tömb visszatérési értékű scriptekben a hivatkozott paraméter tömb típusú (`double[]`, `int[]`, `String[]`, stb.). Például a `p+3` kifejezés csak skalár visszatérési értékű scriptben helyes, míg a `p.length` egy tömb visszatérési értékűben.

Beépített műveletek

Mivel a tick-ek és futások a szimulációs adatok sarokkövei, a MEME rendelkezik ezeket visszaadó, beépített scriptekkel; `$Tick$` és `Run`. A további beépített műveletek listája a következő:

- `get("name")`: visszatér a `name` nevű paraméterrel. A művelet által visszaadott érték és a `name` nevű paraméterből készített változó értéke eltérő lehet:
 - A `get("name")` mindig a paraméter eredeti értékével tér vissza, akkor is ha a `name` nevű globális változó értéke már megváltozott. (Kivéve tömb visszatérési értékű script esetén (mikor a változók tömbök), ha csak egyes elemek értéke változott, nem a teljes tömb.)
 - Ha a megadott név nem érvényes Java azonosító (például `Column0.1`), mivel ilyenkor nincs globális változó, emiatt a paraméter a nevével nem, csak a műveleten keresztül hivatkozható.

Ha nincs `name` nevű paraméter, vagy `null`, a script hibát ad.

- `geti("name")`, `geto("name")`, `getv("name")`: visszatér a `name` nevű input/output/this view jellegű paraméter értékével. Ezen függvények használata akkor szükséges, ha különböző jellegű, de azonos nevű paraméterek vannak. Ekkor a `name` globális változón keresztül, vagy a `get("name")` segítségével csak az első érhető el. A láthatósági sorrend megegyezik a *Wizard* listáinak sorrendjével: *Output*, *This view*, *Input*. Például a `name` nevű *Output* paraméter eltakarja a többi kategória azonos nevű paramétereit.
- `gett("name")`: a technikai paraméterek lekérdezése:
 - `batch`: az aktuális batch sorszáma, vagy `null`, ha az aktuális input sor nem kapcsolódik batch-hez (mert pl. view táblával kapcsolatos)
 - `run`: az aktuális run sorszáma, vagy `null`, ha az aktuális input sor nem kapcsolódik batch-hez
 - `tick`: eredménytábla esetében az aktuális tick, nézet-tábla esetén a sorszám (#)
 - `model`: az aktuális modell neve, vagy `null`, ha az aktuális input sor nem kapcsolódik modellhez
 - `version`: a modellen belül az aktuális verzió neve, vagy `null`, ha az aktuális input sor nem kapcsolódik modellhez

- `starttime`: az aktuális run kezdési időpontja (long), vagy `null`, ha az aktuális input sor nem kapcsolódik eredménytáblához
- `endtime`: az aktuális run befejezési időpontja (long), vagy `null`, ha az aktuális input sor nem kapcsolódik eredménytáblához
- `view`: az aktuális view tábla neve, vagy `null`, ha az aktuális input sor nem kapcsolódik nézet-táblához
- `isgroupfirst`: `true`, amikor csoportosítás van és az aktuális input sor a csoporton belül az első; minden más esetben `false`
- `isgrouplast`: `true`, amikor csoportosítás van és az aktuális input sor a csoporton belül az utolsó; minden más esetben `false`
- `isblockfirst`: `true`, amikor tagolás van és az aktuális input sor a tagon belül az első; minden más esetben `false`
- `isblocklast`: `true`, amikor tagolás van és az aktuális input sor a tagon belül az utolsó; minden más esetben `false`
- `call("fn", args...)`: a `fn` nevű aggregációs művelet meghívása. Pl. `call("AVG", p1, p2)` eredménye a `p1` és `p2` paraméterek értékének átlaga. `args`-ban tetszőleges kifejezések felsorolhatóak, nemcsak paraméterváltozók. Ha 10-nél több van, akkor 1 db `new Object[] { ... }`-ben kell megadni őket. `"fn"` az aggregációs művelet neve, úgy ahogy az *Aggregative operation*: legördülő listában szerepel (a végén lévő „`()`”-al együtt vagy nélküle). Használható a teljes Java osztálynév is (az, amelyik az `ai.aitia.meme.pluginmanager.IVCPlugin` interfészt megvalósítja), pl. az `AVG()` műveletnél `"ai.aitia.meme.builtinvcplugins.Avg"`).