

# Számolótábla

## Deklaratív programozás Gyakorló feladat

2008. szeptember 18.

A számolótábla egy olyan lebutított Excel-tábla, amelyben csak számokat és cellareferenciákat, valamint beépített függvényeket is tartalmazó képleteket engedünk meg. A táblát több modul valósítja meg, közülük csak a `calc.erl` modult adjuk ki forrásnyelven, a többi modulból csak az exportált függvények használatát ismertetjük.

## 1. A modulok leírása

1. A `sheet.erl` modulban található azok a függvények, amelyek szöveges formában megadott adatokkal dolgoznak.

- `sheet:parse/1`: a füzéreként megadott táblázatot strukturált adattá alakítja, amelyen már egyszerű elvégezni a többi műveletet; a füzér első sora a táblázat oszlopszámát tartalmazza, a további sorai pedig `|` (pipe) karakterrel elválasztott cellákat. A `sheet:example(1)` függvény olyan egyszerű példabemenetet ad eredményül, amely szemlélteti a képletek és a referenciák használatát is:

```
"3
 2 | =$a$1+$a$2           | =1+2
 3 | =2*$sum($a$1,$a$3) |
-1 | =3*$b$1             | 5"
```

Ugyanez táblázatos alakban:

	a	b	c
1	2	=\$a\$1+\$a\$2	=1+2
2	3	=2*\$sum(\$a\$1,\$a\$3)	
3	-1	=3*\$b\$1	5

A `sheet:parse(sheet:example(1))` hívás eredménye:

```
[
 [ 2, {"X1+X2", [{"X2", {aa, 1, 2}}, {"X1", {aa, 1, 1}}]}, {"1+2", []}],
 [ 3, {"2*X3", [{"X3", {sum, [{aa, 1, 1}, {aa, 1, 3}]}]}], empty],
 [-1, {"3*X1", [{"X1", {aa, 2, 1}}]}], 5 ]
```

- `sheet:eval/2`: egy füzéreként megadott Erlang-kifejezést a mellékelt változó-behelyettesítések figyelembe vételével értékeli ki, pl.

```
sheet:eval("A+B*2", [{"A", 5}, {"B", 3}]) => 11
```

2. A `calc.erl` modul tartalmazza a számolótáblát kiértékelő függvényeket.

- `calc:process/1`: a megadott táblában kiértékel minden olyan kifejezést, amely nem függ más cellától, vagy csak kiértékelt cellákra mutató referenciákat tartalmaz:

```
calc:process(sheet:parse(sheet:example(1))). =>
[[2,5,3],[3,8,empty],[-1,{"3*X1",[{'X1',{aa,2,1}}]},5]]
```

Itt a b3-as cellát még nem lehetett kiértékelni, mert a bemenetben a b1-es cella képlet volt, nem szám.

- `calc:processfix/1`: a `calc:process/1` függvényt addig hívja meg ismételten, amíg van feloldható referencia. Ha körkörös hivatkozás van a táblában, az érintett cellákba a `loop` atomot rakja. Az előző példa eredménye:

```
calc:processfix(sheet:parse(sheet:example(1))). =>
[[2,5,3],[3,8,empty],[-1,15,5]]
```

3. A `func.erl` modul tartalmazza a speciális számológépa-függvényeket (l. `<funcall>` a formális leírásban) megvalósító függvényeket.

Jelenleg egyetlen ilyen függvény van: `$$sum`, amely a paramétereként megadott cellatartomány tartalmát összegzi.

## 2. Gyakorló kifeladatok

A gyakorlaton először az alább felsorolt kifeladatok közül fogunk megoldani néhányat, majd elemezni fogjuk a `calc.erl` modult. A kifeladatok egy részét meghagyjuk otthoni gyakorlásra.

### 1. Lista kezelése

- Elem beszúrása az  $n$ . helyre, az  $n$ . elem elhagyása
- Hozzárendelés megvalósítása párok listájával, hozzárendelt érték kikeresése
- Rekurzív, ill. listanézetes megoldások megírása, összevetése
- Lista szétszedése két részre az első rész hosszának megadásával
- Listák bezippzárítása, kizippzárítása

### 2. Mátrix kezelése

- Adott cella, sor, oszlop, blokk elővétele
- Adott sor, oszlop, blokk elemeinek összeadása, átlagolása
- Adott cella, sor, oszlop, blokk tartalmának törlése
- Adott cella, sor, oszlop, blokk tartalmának másolása cellába, sorba, oszlopba, blokkba

### 3. Részletes leírás

#### 3.1. A bemenet szintaxisa

A tábla bármely cellájának tartalma a következő lehet:

```
<cellcontent> ::= <empty> | <number> | ' = ' <expr>
<empty> ::= ' ' % whitespace, results in error if evaluated
<number> ::= <integer> | <float> % Erlang built-in types
<expr> ::= % valid Erlang expression without variables; extra terms allowed: <cellref>, <funcall>
<cellref> ::= <refmark> <colnum> <refmark> <rownum>
<refmark> ::= ' $ ' | ' ! ' % absolute and relative reference, respectively
<colnum> ::= % a lower case letter denoting columns 1 to 26
<rownum> ::= % an integer between 1 and the number of rows
<funcall> ::= ' $ $ ' <funapp>
<funapp> ::= <funname> ' ( ' <args> ' ) '
<funname> ::= % a sequence of lower case letters
<args> ::= <args> ' , ' <arg> | <arg>
<arg> ::= <cellref>
```

Megjegyzések:

- A felülvessző ( ' ) a terminális szimbólumokat határoolja.
- A % a megjegyzés, szöveges magyarázat jele.
- Az abszolút és a relatív hivatkozás között az a különbség, hogy a belső ábrázolásban a relatív hivatkozás az eredeti cellapozíciótól számított különbséget tárolja, így a cella tartalmának másolásakor változik a jelentése, ellentétben az abszolút hivatkozással.

#### 3.2. A sheet.erl modul

A sheet.erl modul a sheet:eval/2, sheet:parse/1, sheet:indexlist/1 függvényeket exportálja.

A sheet:parse/1 függvény paramétere a füzéreként ábrázolt számolótábla. Az első sorban a számolótábla oszlopainak a számát, a többi sorban a számolótábla egy-egy sorát kell megadni. A soron belül a cellákat a | (pipe) jellel kell elválasztani egymástól; ha az oszlopszámnál kevesebb cellát adunk meg, a hiányzó cellákat a program üresnek veszi, ha többet adunk meg, a felesleges cellákat elhagyja. Fontos, hogy az üres cellákba legalább egy szóköz kerüljön, hacsak nem a sor végén vannak. A cellák lehetséges tartalmát az előző szakaszban specifikáltuk.

A sheet:parse/1 függvény eredménye egy olyan listák listája, amely az egyes cellák tartalmát a következő módon ábrázolja:

szám	szám (egész vagy lebegőpontos)
üres cella	empty atom
egyenlőségjellel bevezetett képlet	{Keplet, Behelyettesitesek}, ahol Keplet a képlet szöveges változata, melyben a hivatkozásokat és a beépített függvényeket változónév helyettesíti, Behelyettesitesek pedig párok listája, amely a képletben található változónevekhez az eredeti tartalmukat rendeli: <ul style="list-style-type: none"><li>• referencia → {&lt;aa ar ra rr&gt;, Oszlop, Sor}</li><li>• függvényhívás → {Fuggvenynev, Argumentumlista}</li></ul>
minden egyéb esetben	error atom

A `calc:process/1` és `calc:processfix/1` függvények ebben az alakban fogadják a számolótáblát. Az `aa`, `ar`, `ra` és `rr` atomok az oszlop- és sorszám abszolút, illetve relatív voltát jelölik.

A `sheet:eval/2` egy füzéreként megadott Erlang-kifejezést értékeli ki a második argumentumában átadott változó-behelyettesítések figyelembe vételével.

A `sheet:indexlist/1` függvény paramétere egy tetszőleges `Xs` lista, az eredménye pedig egy olyan `{X, I}` párokból álló lista, ahol az `X` az `Xs` `I`-edik eleme.

### 3.3. A `calc.erl` modul

A `calc.erl` modul függvényei kiértékelik a számolótáblát, azaz a cellák tartalmát egyszerűbb, illetve a lehető legegyszerűbb (ún. kanonikus) kifejezéssé alakítják. A függvényalkalmazások eredménye az egyszerűsített tábla.

A `calc.erl` program a következő függvényeket használja:

`lists:all/2`, `lists:any/2`, `lists:nth/2`.

Javasoljuk a `lists` modul többi függvényének tanulmányozását is, különösen a következőket:

`lists:append/1`, `lists:append/2`, `lists:dropwhile/2`, `lists:duplicate/2`,  
`lists:filter/2`, `lists:flatten/1`, `lists:flatten/2`, `lists:foldl/3`,  
`lists:foldr/3`, `lists:keysearch/3`, `lists:last/1`, `lists:map/2`, `lists:member/2`,  
`lists:reverse/1`, `lists:reverse/2`, `lists:seq/3`, `lists:sort/1`, `lists:split/2`,  
`lists:splitwith/2`, `lists:sublist/2`, `lists:sublist/3`, `lists:subtract/2`,  
`lists:sum/1`, `lists:takewhile/2`, `lists:unzip/1`, `lists:zip/2`.

A gyakorlatra való felkészülésként az Erlang/OTP leírásban (pl. <http://erlang.org/doc/index.html>, Basic Applications, stdlib könyvtár) meg kell nézni a függvények leírását.

A `calc.erl` modul a `calc:process/1`, `calc:processfix/1` függvényeket exportálja.

A `calc:process/1` függvény egyszeri meghívása kiértékeli a táblát, de csak a közvetlen cellahivatkozásokat oldja fel. Ha a létrejött új cellahivatkozásokat is fel akarjuk oldani, a függvényt ismét meg kell hívni. Ezt teszi a `calc:processfix/1` függvény, amely a nem körkörös összes cellahivatkozást feloldja, a körkörös hivatkozásokat tartalmazó cellákba pedig a `loop` atomot írja.

### 3.4. A `func.erl` modul

A `func.erl` modul a `func:sum/3` függvényt exportálja.

A függvény paraméterezése: `func:sum({Sheet, CColNum, CRowNum}, RefTL, RefBR)`, ahol

<code>Sheet</code>	A teljes táblázat
<code>CColNum</code>	Az aktuális cella oszlopa
<code>CRowNum</code>	Az aktuális cella sora
<code>RefTL</code>	Az összegzendő blokk bal felső sarkára hivatkozó referencia
<code>RefBR</code>	Az összegzendő blokk jobb alsó sarkára hivatkozó referencia