

Smalltalk 1.

Alapok

- tisztán OO nyelv, MINDEN objektum
- függvények helyett üzenetek
- az objektumokban definiálható a viselkedés az egyes üzenetekre
- ha a küldött üzenetre nem definiált ez, akkor olyan, mintha a `doesNotUnderstand` üzenetet küldenénk, ami alpból általában egy debug üzenetet ír ki, de felül is definiálható

GNU Smalltalk interpreter (pubon is fent van)

- értelmező: soronként kell gépelni a programot, amit szakaszonként értelmez és végrehajt

- **szakasz vége: !**
- **utasítás vége: .**

Használat:

- *gst* - értelmező indítása
- kilépés: Ctrl+D
- ***gst -q*** - quiet mód, nem ír ki annyi mindent
- *gst fejl.st* - a megadott fájlból beolvassa az utasításokat, majd kilép
- *gst fejl.img* - a megadott fájlból beolvas egy elmentett snapshotot

(Snapshot elmentése:

Smalltalk snapshot: 'valami.img'

ObjectMemory snapshot: 'valami.img')

Üzenetek

- *Unáris*
 - Objektum üzenet, ezt írja ki először,
 - pl. ***odd, negated, asString, print, displayNil***
sqrt (négyzetgyökvonás)
- *Bináris*
 - Pl.: aritmetikai műveletek, ***5+3, max***
- *Kulcsszavas*
 - Több paraméteres függvényt helyettesítünk
 - ***obj kulcs_1:arg_1 ... kulcs_n:arg_n***

Precedencia

- $(5 + 3 * 4)$ *printNI!*

Eredmény: 32, mert nem szorozni fog először. Zárójelezni kell!

- $7*2$ *printNI!* → 2

először kiírja a 2-t, utána szoroz 7-tel.

Helyesen: $(7*2)$ *printNI!*

- $(2+5)$ *negated printNI!*

Eredmény: -3, mert $2+(-5)=-3$.

- $(2 + 3 \text{ max: } 3 + 4)$ *printNI.* → 7

- $(\$A \text{ max: } \$B)$ *printNI.* → $\$B$

Változók

- **Lokális változó:** csak az adott szakaszon belül él.
 - `|x y z|` - deklaráció (2 pipeline között)
 - `x := 2.` (egyszeres értékadás)
 - `x := y := z := 2.` (többszörös értékadás)
- **Globális változó:**
 - Smalltalk at: #változonev put: érték !*

Kiíratás

- ***print***: kiírat, sztringnél ír aposztrófot
- ***printNL***: kiírat és új sorba ugrik, sztringél ír aposztrófot
- ***display***: kiírat, sztringél nem ír aposztrófot
- ***displayNL***: kiírat és új sorba ugrik, sztringél nem ír aposztrófot

- ***[Transcript show: ' valami '.]*** : ~ print

- Példák kiíratásra:
 - 5 printNI.
 - \$B printNI. és \$B displayNI.
 - 'alma' printNI. és 'alma' displayNI!
 - 5 odd printNI. → true
 - 4 odd printNI. → false

'Hello World' printNl!

vagy:

Smalltalk at: #hello put: 'Hello World!' !

hello printNl.

Komment: idézőjelek közé írjuk, több soros.

”ez egy komment Smalltalkban”

Beolvasás

- *x := stdin nextLine.*
- A standard input a billentyűzet.
- Springként olvas be! pl.:

x := stdin nextLine.

(prompt...) *alma* (Enter)

x printNl. → 'alma'

y := stdin nextLine.

(prompt...) *5* (Enter)

y printNl. → '5'

Aritmetikai műveletek

- + - összeadás
- - - kivonás
- * - szorzás
- / - tört osztás
- // - egész osztás hányadosa
- \\ - egész osztás maradéka

pl.: $(2/3)$ *printNI!* → 2/3

$(4/6)$ *printNI!* → 2/3

$((2/3)*5)$ *printNI!* → 10/3

$(2/3.0)$ *printNI!* → 0,666

Karakter üzenetek

- ***Character space*** - szóköz karakter
- ***Character eof*** - End Of File karakter (fájl vége)
- ***Character nl*** - new line kar. (új sor)
- ***\$a asUppercase*** - nagybetűvé konvertálás
- ***\$A asLowercase*** - kisbetűvé konvertálás
- ***\$a isLetter*** - betű-e a karakter
- ***\$1 isDigit*** - számjegy-e a kar.
- ***\$1 digitValue*** - (számjegy)karakter számértékét adja vissza
- ***\$a isSeparator*** - elválasztó karakter-e
- ***\$a asciiValue*** - ASCII kódot ad vissza

Integer üzenetek

- 5 *odd* - páratlan-e
- 4 *even* - páros-e
- 5 *negated* - szám ellentettje
- 3 *factorial* - faktoriális
- 5 *squared* - négyzetre emelés
- 5 *raisedTo: 3* - hatványozás
- 12 *gcd: 20* - legnagyobb közös osztó
- 60 *lcm: 18* - legkisebb közös többszörös
- 2+3 *min: 3+4* - minimum
- 2+3 *max: 3+4* - maximum
- 66 *asCharacter* - ASCII kódot karakterré alakít

Az én házim: type casting, type conversion

- A ST-ban konverzió van (nincs kasztolás).
- Formája:

Vált1 := vált2 asXxx .

Ahol vált1 és vált2 lehet ugyanaz a változó is.

Xxx : String, Integer, Float

- Integer → Float – 2 → 2.0
- Integer → String – 2 → '2'
- String → Integer – '2' → 2, '2,9' → 2 , '2alma' → 2, 'alma2' → 0, 'alma'
→ 0
- Float → Integer – működik, egész számra kerekít: 2.5 → 3
- Float → String – 2.449 → '2.449'
- String → String, Integer → Integer, Float → Float
- Illetve: (Integer → Character – ASCII kódot karakterré alakít)

Jövő heti tervek:

- Blokkok
- Vezérlési szerkezetek
- Objektumok összehasonlítása
- Kollektciók
- Műveletek kollektciókkal