# Celldown: a textual spreadsheet language or: "code as Ul"

Tijs van der Storm, CWI @tvdstorm / storm@cwi.nl

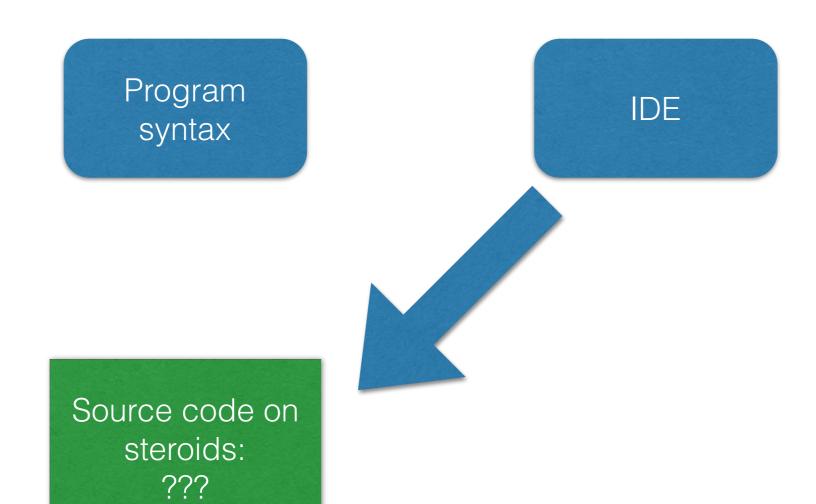


## Language interaction design

- Explore the connection between language and IDE
- Lorentz Workshop in 2013
- Hypothesis: IDE and language can/should not be separated.

Program syntax IDE

Spreadsheets
Jetbrains MPS
"Mendix"
Intentional



## Hyperbole of the day

## Source code all the things!

# Background



#### Copy-Paste Tracking: Fixing Spreadsheets Without Breaking Them

Felienne Hermans
Delft University of Technology
f.f.j.hermans@tudelft.nl

Tijs van der Storm Centrum Wiskunde & Informatica storm@cwi.nl



#### Enter a formula in D2

Α	В	С	D
ld	Lab	Exam	Final
s123	6	7.5	=(B2+C2)/2
s456	9	7	
s789	5	3.5	



#### Copy it to D3:D4

Α	В	С	D	
ld	Lab	Exam	Final	
s123	6	7.5	=(B2+C2)/2	4
s456	9	7	=(B3+C3)/2	) origin
s789	5	3.5		



#### Edit the "copy" in D4

Α	В	С	D
ld	Lab	Exam	Final
s123	6	7.5	=(B2+C2)/2
s456	9	7	=(B3+C3)/2
s789	5	3.5	=round((B4+C4)/2)



#### Reconcile D2:D3 with the edits in D4

Α	В	С	D
ld	Lab	Exam	Final
s123	6	7.5	=round((B2+C2)/2)
s456	9	7	=round((B3+C3)/2)
s789	5	3.5	=round((B4+C4)/2)

"Copy" (many)	"Source" (one)
Reference	Declaration
Stack frame	Procedure call
Inlining	Procedure
Text output	Template
Object	Class
Styled element	Style sheet
View	Database table
Unrolling	Loop

#### "abstraction without indirection"



Web Images Videos News More ▼ Search tools

5 results (0.19 seconds)

#### [DOC] 4. towards a theory of copying - Subtext

www.subtext-lang.org/OOPSLA05.doc ▼

by J Edwards - Cited by 58 - Related articles

This principle is called **abstraction without indirection**. It is made possible by the ability of the program representation to automatically react to change.



## We don't need no stinking GUI

- Rascal language workbench
- "Dealers of syntax"

## Celldown

```
table grades =
# A / B / C / D
1: Lab | Exam | Avg | Grade
2: 7 | 4 | = (A2 + B2) / 2 | = round(C2)
3: 3 | 6 | = (A3 + B3) / 2 | = round(C3)
4: 9 | 10 | = (A4 + B4) / 2 | = round(C4).
```

### Demo

```
table grades = # A / B / C / D
                                                                 / E
                   1: | Lab | Exam | Avg | Grade
                   2: | 7 | 7 | = (B2 + B3) / 2 | = round(D2)

3: | 3 | 7 | = (B3 + C3) / 2 | = round(D3)

4: | 9 | 10 | = (B4 + C4) / 2 | = round(D4).
view grades = # A / B / C / D / E
                  1: | Lab | Exam | Avg | Grade
                 2: | 7 | 7 | 5. | 5. | 3: | 3 | 7 | 5. | 5. | 4: | 9 | 10 | 9.5 | 10...
test grades E2 * 2 == B2 + C2 expected 14., got 10.
repl for grades
> A2 + B2
=> 7<sub>•</sub>0
>B2 + B2
=> 7.0
```

>

## Source code?

- Views: tests, examples, stacks, explanations
- Editing gestures: copy-paste, structure editing
- Transformations: refactoring, inlining, expansion
- Commands: break points, watches, probes

•

We can do much more with source code than just write and document our programs

# "Syntax all the way down"

Or, perhaps more accurately: Syntax all the way up

/ht Roly Perera

# Source code all the things!