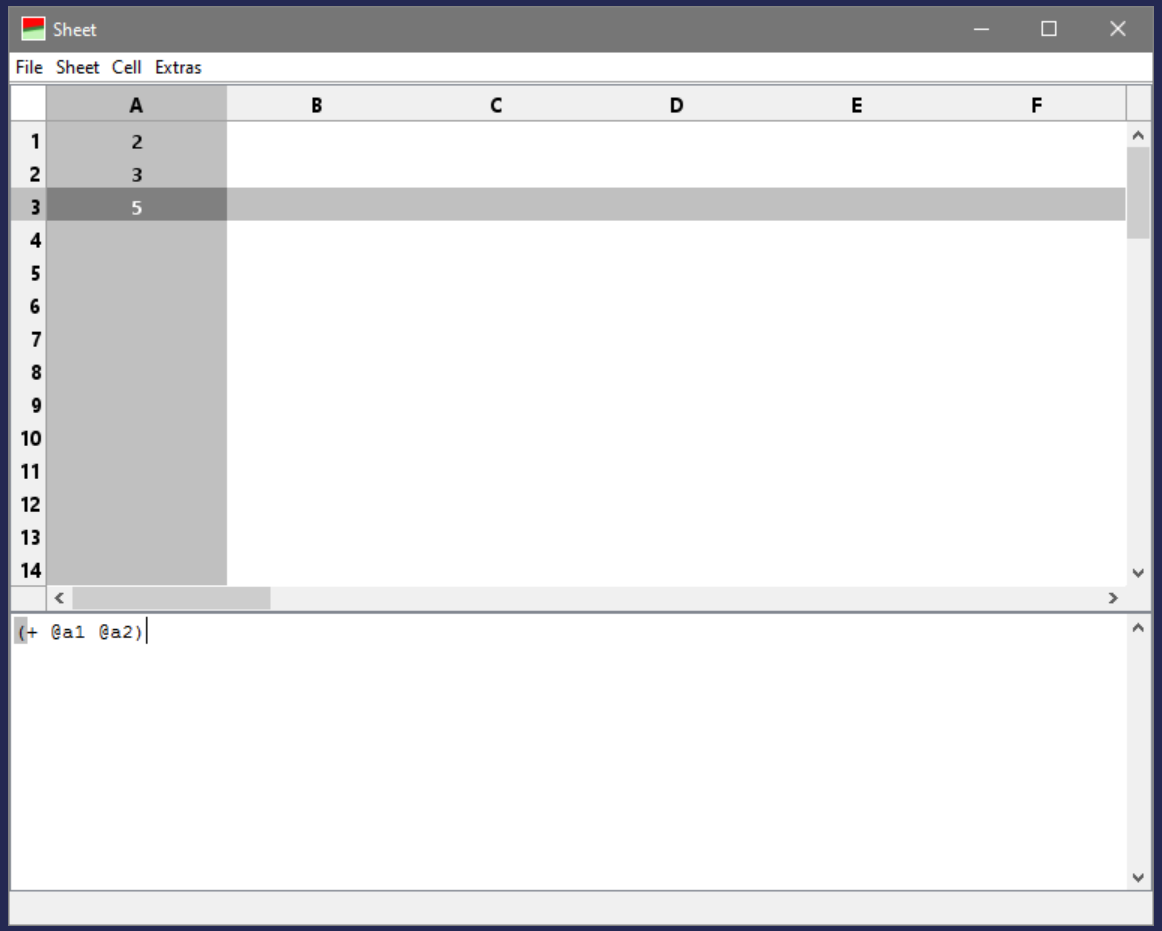
Interactive Programming with a modern Lisp Dialect

Abstract

- Most computer users are familiar with spreadsheets, for example Excel.
- An important feature of spreadsheets is interactivity – a value in one cell is changed and other dependent cells are recalculated instantly.
- Spreadsheets can be used in a programming context. They are an interactive alternative to read-eval-print loops or to shells.
- Spreadsheets especially fit well to functional programming.
- The standardization of Common Lisp 1994 almost stopped Lisps evolution. Since then other programming languages incorporated new features, that are missing in Lisp.
- Many interesting features of languages like Ruby, Go or X10 can be included into Lisp elegantly.
- A spreadsheet that uses Lisp for formulas can be used like any other spreadsheet and also for programming or mathematics education, prototyping or code generation. And it is fun to use if you like parentheses :-)

Spreadsheet Basics



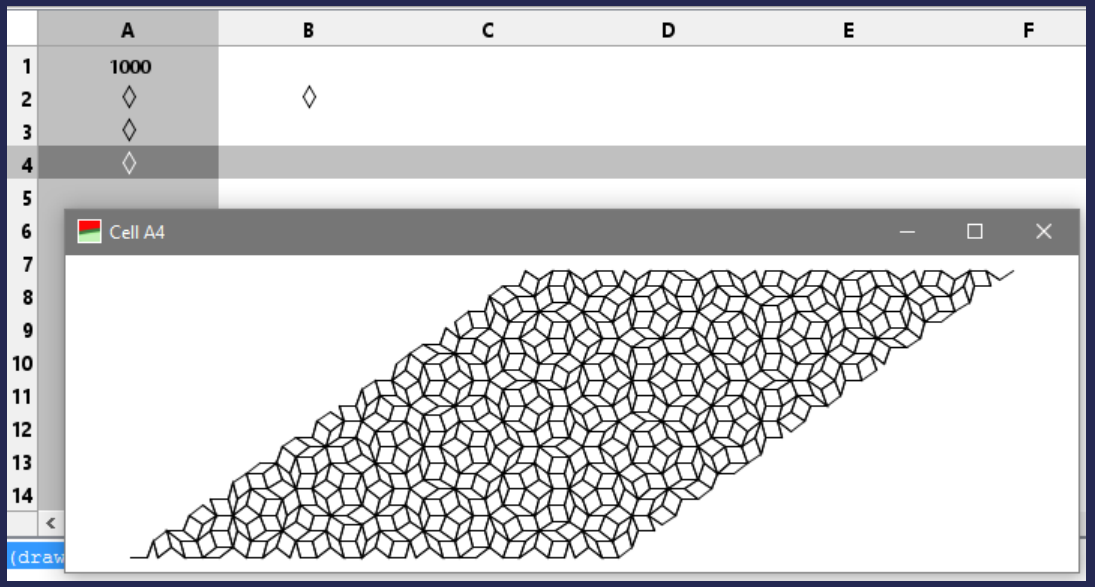
The upper part of the window shows the cell values of rows 1 to 14 and columns A to F. Cell A3 is selected. The formula of the selected cell is shown below.

Cells contain Lisp **S-Expressions**. That can be simple numbers like 2 and 3 in cells A1 and A2 or a formula like (+ @a1 @a2) in cell A3.

The **@**-sign followed by cell coordinates is used for **referencing cell values** in formulas.

Remember that Lisp consequently uses **polish notation**.

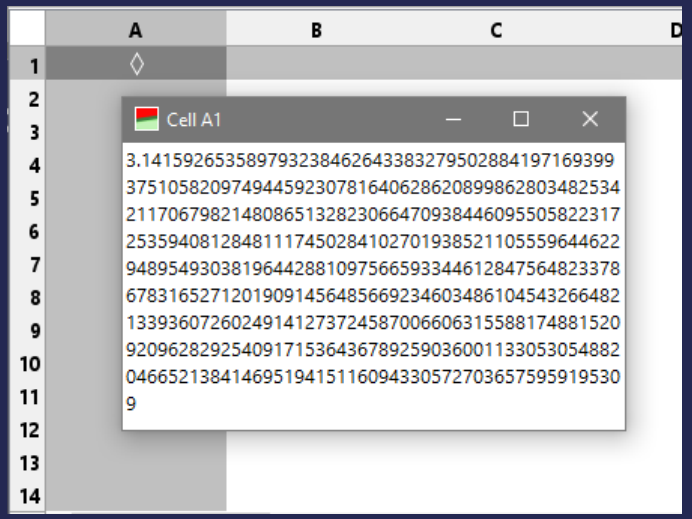
Cell Values



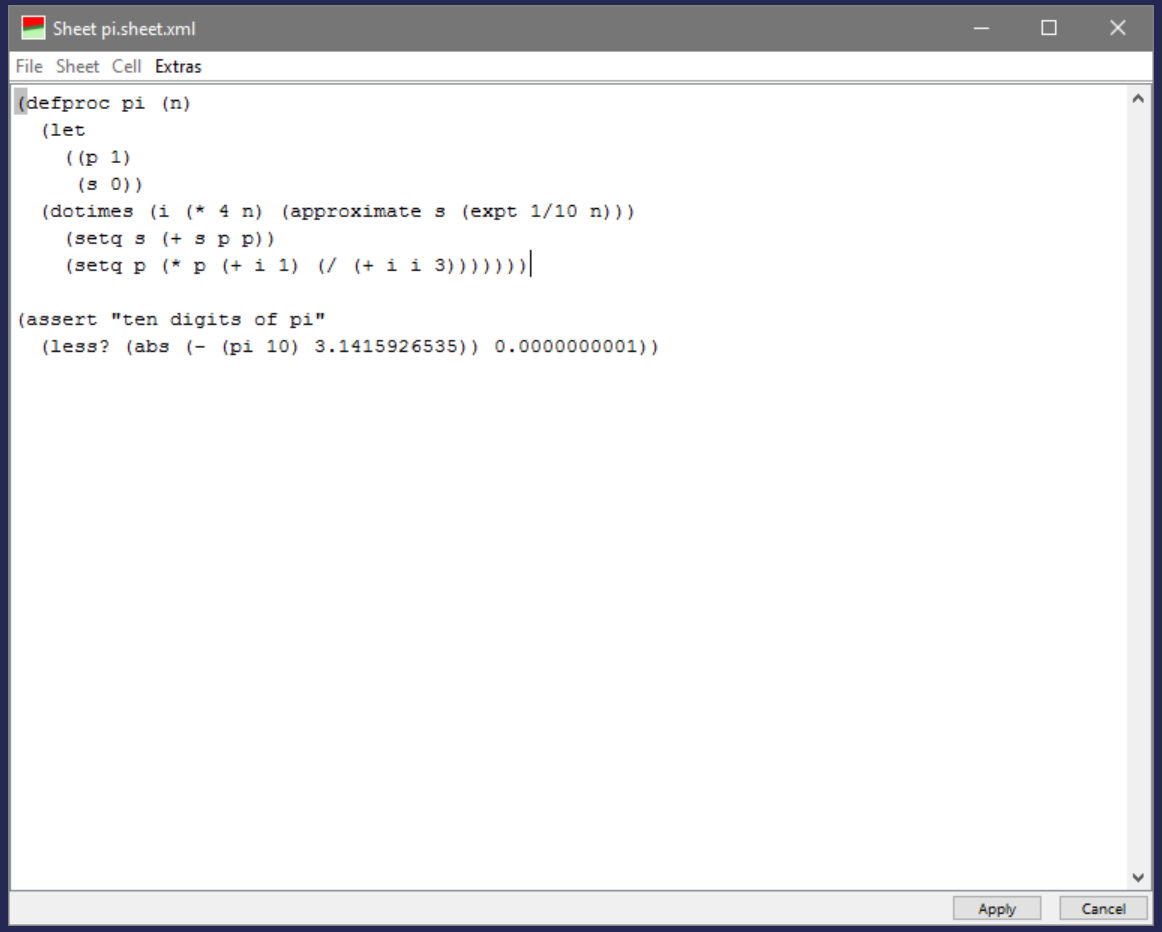
If cell values are graphics (as in this example a Penrose tiling), the menu command **Cell > Magnify** or **Ctrl+M** opens a separate window that displays them.

The menu command acts on the selected cell.

Also if the textual representation of a cell value is too long, a rhombus is shown in the main window. Using the menu command **Cell > Magnify** opens a separate window with the cell value – here 500 digits of Pi.



Programs



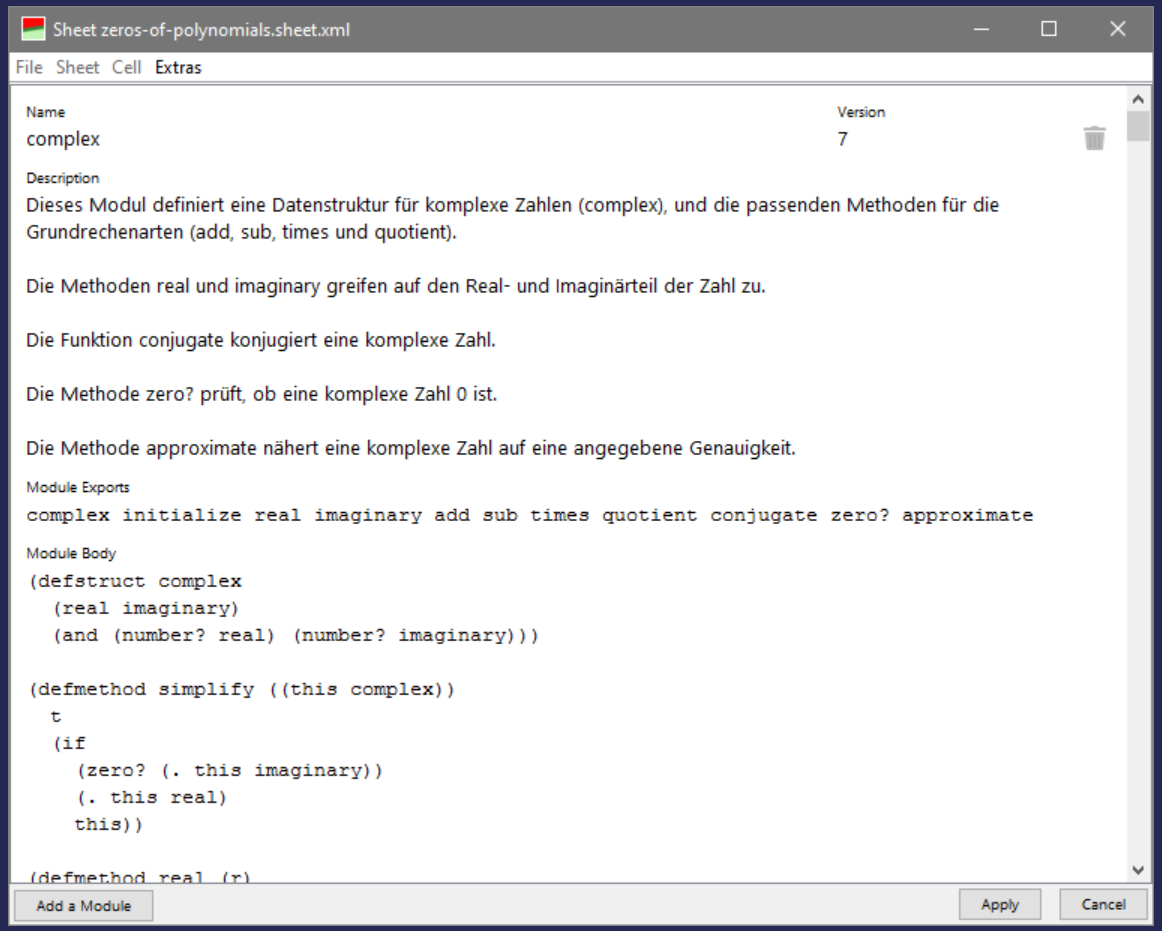
The menu command **Sheet > Show Program ...** displays the program editor.

Programs typically define functions, methods or classes (with **defproc**, **defmethod** or **defclass**). These can be used in the formulas of the cells.

Programs can also have unit tests (with **assert**).

The shown program calculates Pi. It is based on the spigot algorithm of Stanley Rabinowitz and Stan Wagon.

Modules



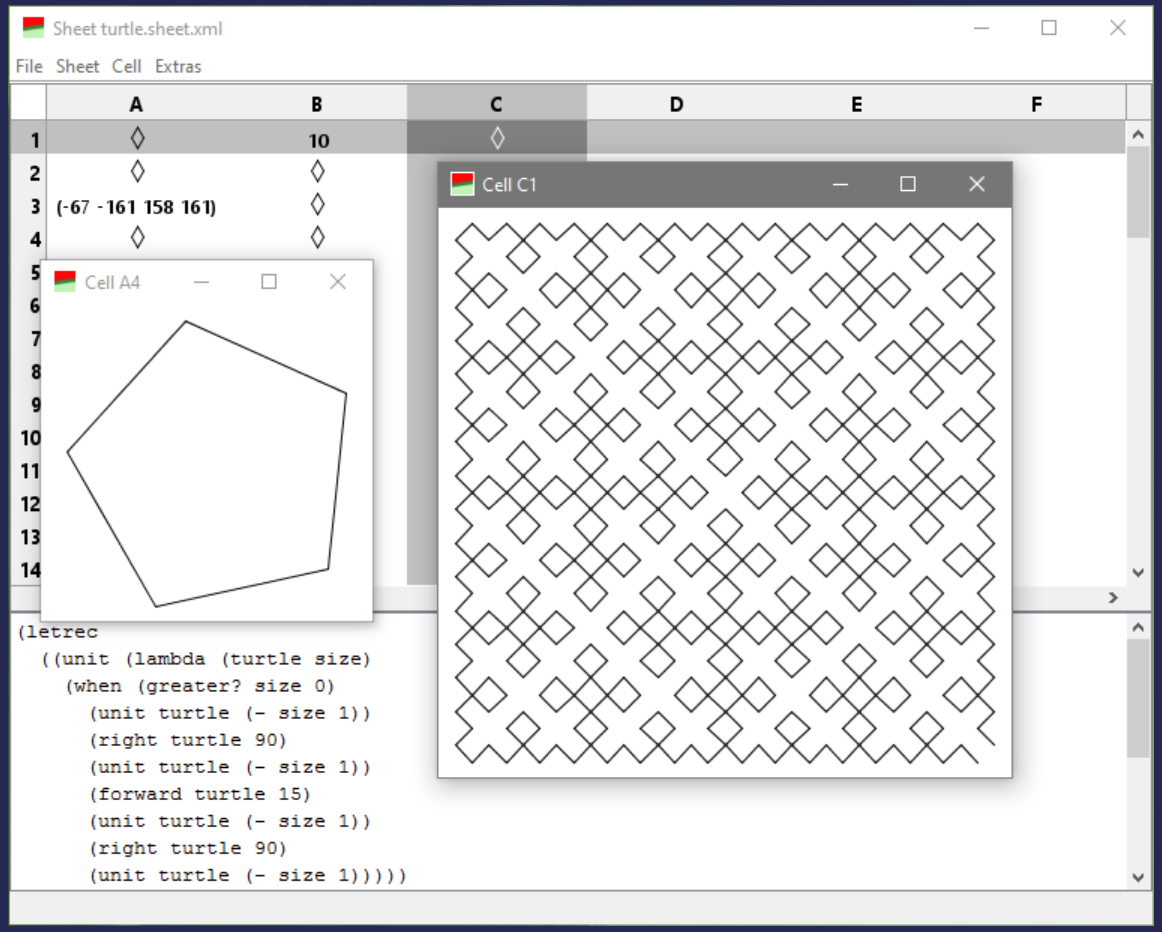
The menu command **Sheet > Show Modules ...** displays the module editor.

Modules are reusable programs with a **name**, a **version**, a **description** (sorry, only german in the shown example) and a list of accessible (exported) **definitions**.

These definitions can be used in the program and in the formulas.

The shown module defines a structure (that is a simple class) and functions for complex numbers.

Example – Turtle Graphics



The shown sheet contains the picture of a pentagram in cell A4 and a recursive space-filling curve in cell C1.

The turtle graphics module defines the class turtle.

Instances of this class represent turtles that sit on a piece of paper and hold a pen.

The turtles can be commanded by method calls e.g. „pen down“, „forward 10 steps“, „turn right 90 degrees“ etc.

More Examples

More examples can be found in the functional programming tutorial (see the link on the last slide) - currently in german language only.

Recursion
Collatz Problem
Palindromes
Horner Scheme
Shoelace Formula
Calculate Pi
Towers of Hanoi
Pascal's Triangle
Newton's Method
Regula Falsi
Bisection Method
Halley's Method
Polynomial Ring
Complex Numbers
Polar Coordinates
Roots of Polynomials
Linear Algebra

Laplace's Formula
Cramer's Rule
Circle through 3 Points
Central Projection
Stars
Square Limit
Turtle Graphics
Lindenmayer System
Dragon Curve
Penrose Tiling
Search in Graphs
Depth-first Search
Breadth-first Search
Water Jug Problem
Pairing Heaps
Best-first Search
A* Search

8 Puzzle
Disjoint Set Data Structure
Kruskal's Algorithm
Terms
Formulas
Horn Clauses
Unification
Resolution
Sieve of Eratosthenes
Dining Philosophers
Distributed Sorting
Residue Class Ring
Elliptic Curves
Factorization
Distributed Factorization
Parsing
Translation of „Adaish"

Added Language Features

Since the Common Lisp standard was finished in 1994 new programming language features were invented. Also computer hardware changed. This lead to the following additions for the Lisp dialect used here.

- **Arguments** for function calls are **evaluated concurrently**, if enough processor cores are available. In any case the order of argument evaluation (e.g. left to right) is not defined. Some special forms (e.g. if, and, progn) are exceptions to this general rule.
- The way **Ruby** deals with **classes** and **instances** fits better to Lisp as CLOS (the Common Lisp Object System). So classes do not define the data members of their instances. Classes are only used to specify (multiple) dispatch for methods of generic functions.
- The methods of **generic functions** need not have all the same **arity** as in CLOS.
- Methods allow **predicate dispatch**.
- **Channels** and **Goroutines** from Go – go, await-future, make-channel, receive-from-channels, send-on-channel, close-channel, closed-channel?
- **Places** and **remote execution** from X10 – get-places, at

Download and Installation

You can download the spreadsheet application with the link „Calc.jar“ from the page http://simplysomethings.de/functional+programming/calc.html.

The JAR-File requires a version of Java 8 installed.

You can start the spreadsheet with a double click on the JAR-File or with `java -jar Calc.jar` on the command line.

The source code is included in the JAR-File. Noncommercial use is free. You are not allowed to sell the program without the authors written permit. The Artistic License applies, see https://en.wikipedia.org/wiki/Artistic_License.

For a full documentation of the Lisp dialect used here, please visit http://simplysomethings.de/lisp+package/index.html.

Links

- History of Spreadsheets
  https://en.wikipedia.org/wiki/Spreadsheet#History
- Common Lisp
  https://en.wikipedia.org/wiki/Common_Lisp
- Polish notation
  https://en.wikipedia.org/wiki/Polish_notation
- A Spigot Algorithm for the Digits of Pi
  http://www.mathpropress.com/stan/bibliography/spigot.pdf
- Turtle Graphics
  https://en.wikipedia.org/wiki/Turtle_graphics
- Functional Programming Tutorial
  http://simplysomethings.de/functional+programming/tutorial.html
- Predicate Dispatch
  https://en.wikipedia.org/wiki/Predicate_dispatch
- Places and the at Construct in X10
  http://x10.sourceforge.net/documentation/intro/latest/html/node4.html