Literate Programming and Finances

Vinícius Gajo

[2022-11-02 qua]

Contents

1 Introduction

> Literate programming is a programming paradigm introduced in 1984 by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed (embedded) with snippets of macros and traditional source code, from which compilable source code can be generated. The approach is used in scientific computing and in data science routinely for reproducible research and open access purposes. Literate programming tools are used by millions of programmers today.
>
> — [1]

This repository holds an example of literate programming used in the finances landscape. The macros used in the program are written in F#, which is a functional-first, general purpose, strongly typed, sister language of C# and other .NET implementations.

1.1 Setup

When creating this example I used those software versions:

- GNU Emacs 28.2.
- The `.emacs` configuration mentioned in this repository release: link.

Disclosure: some configurations will not work, even if you have the same `.emacs` file since it will depend on external packages (LaTeX specific packages for example), and the file system structure of your computer.

2 The problem

In this example, we're going to tackle an investment problem, which could be stated as:

- Suppose you're going to invest your money into an application that gives you "h%" of this money each month, as long as you keep it in this investment. During the time you keep it there, you can also add more money (in a month basis), and this money will follow the same rule stated before, but considering that it will start producing more only in the next month that you added it.

To make it more clear, let's use some mathematical notation.

Consider that in the first month, the money you have (z_0) is only the initial quantity you decided to invest (x_0).

$$z_0 = x_0 \tag{1}$$

Then, in the second month, the initial money will increase by a quantity given by h and you're going to increment the value adding y_0. So, in the second month, your money will be:

$$\begin{aligned} z_1 &= z_0 \times h + y \\ &= x_0 \times h + y \end{aligned} \tag{2}$$

In the third month you repeat the same operation. This time, your money will be:

$$\begin{aligned} z_2 &= z_1 \times h + y \\ &= (x_0 \times h + y) \times h + y \\ &= (x_0 \times h^2) + (y \times h) + y \end{aligned} \tag{3}$$

And the following months you keep doing the same operation, until the month n. By the n-month your money will be:

$$\begin{aligned} z_n &= z_{n-1} \times h + y \\ &= (x_0 \times h^n) + (y \times h^{n-1}) + (y \times h^{n-2}) + ... + (y \times h) + y \end{aligned} \tag{4}$$

3 References

[1] - https://en.wikipedia.org/wiki/Literate_programming